BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

02 233
ARC

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００３年　３月２５日

出願番号
Application Number: 特願２００３－０８３２４３

[ST.10/C]: [ＪＰ２００３－０８３２４３]

出願人
Applicant(s): インターナショナル・ビジネス・マシーンズ・コーポレーション

２００３年　５月２０日

特許庁長官
Commissioner,
Japan Patent Office 太田信一郎

出証番号　出証特２００３－３０３７０４３

【書類名】　特許願

【整理番号】　JP9020233

【提出日】　平成15年 3月25日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 12/00

G06F 17/30

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　工藤　道治

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　村田　真

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　戸澤　晶彦

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　羽田　知史

【特許出願人】

【識別番号】　390009531

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【弁理士】

【氏名又は名称】　坂口　博

【代理人】

【識別番号】　100091568

【弁理士】

【氏名又は名称】　市位　嘉宏

【代理人】

【識別番号】　100108501

【弁理士】

【氏名又は名称】　上野　剛史

【復代理人】

【識別番号】　100104880

【弁理士】

【氏名又は名称】　古部　次郎

【選任した復代理人】

【識別番号】　100118201

【弁理士】

【氏名又は名称】　千田　武

【手数料の表示】

【予納台帳番号】　081504

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9706050

【包括委任状番号】　9704733

【包括委任状番号】　0207860

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　情報処理装置、データベース検索システム及びそのアクセス権解析方法

【特許請求の範囲】

【請求項１】　構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析する情報処理装置において、

前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部と、

アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部と、

前記検索オートマトン生成部にて生成された前記検索オートマトンおよび前記アクセス制御オートマトン生成部にて生成された前記アクセス制御オートマトンに関する論理演算を行って、前記パス式によるデータベース検索におけるアクセス権を判定する論理演算部と

を備えることを特徴とする情報処理装置。

【請求項２】　前記データベースに格納されているデータファイルの構造を示すスキーマからスキーマオートマトンを生成するスキーマオートマトン生成部をさらに備え、

前記論理演算部は、前記スキーマオートマトン生成部にて生成された前記スキーマオートマトンを考慮して前記アクセス権の判定を行うことを特徴とする請求項１に記載の情報処理装置。

【請求項３】　前記データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理部をさらに備え、

前記スキーマオートマトン生成部は、前記パステーブル管理部に管理されている前記パステーブルから前記スキーマオートマトンを生成することを特徴とする請求項２に記載の情報処理装置。

【請求項４】　前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出部をさらに備えることを特徴とする請求項１に記載の情報処理装置。

【請求項５】　前記問い合わせ式から抽出された個々の前記パス式に対する前記論理演算部によるアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定部をさらに備えることを特徴とする請求項４に記載の情報処理装置。

【請求項６】　構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析する情報処理装置において、

前記データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理部と、

前記データベースに対する検索条件を記述したパス式によって前記パステーブル管理部に管理されている前記パステーブル中の所定のパスを選択し、アクセス制御規則を記述したアクセス制御ポリシーを適用して、当該所定のパスに対する当該パス式によるデータベース検索におけるアクセス権を判定するアクセス権判定部と

を備えることを特徴とする情報処理装置。

【請求項７】　前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部と、

前記アクセス制御規則を記述した前記アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部とをさらに備え、

前記アクセス権判定部は、前記検索オートマトン生成部にて生成された前記検索オートマトンを用いて前記所定のパスの選択を行い、前記アクセス制御オートマトン生成部にて生成された前記アクセス制御オートマトンを用いて前記所定のパスに対するアクセス権の判定を行うことを特徴とする請求項６に記載の情報処理装置。

【請求項８】　前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出部をさらに備えることを特徴とする請求項６に記載の情報処理装置。

【請求項９】　前記問い合わせ式から抽出された個々の前記パス式に対する前記アクセス権判定部によるアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アク

セス権判定部をさらに備えることを特徴とする請求項８に記載の情報処理装置。

【請求項１０】　ＸＭＬ文書を格納したデータベースと、

前記データベースに対する検索に用いられる検索条件を記述したパス式およびアクセス制御規則を記述したアクセス制御ポリシーに基づいて、当該パス式を用いた前記データベースの検索におけるアクセス権が、常に許可、常に拒否、不確定のいずれであるかを判定するアクセス権解析器と

を備えたことを特徴とするデータベース検索システム。

【請求項１１】　前記アクセス権解析器は、

前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部と、

前記アクセス制御規則を記述した前記アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部と、

前記検索オートマトン生成部にて生成された前記検索オートマトンおよび前記アクセス制御オートマトン生成部にて生成された前記アクセス制御オートマトンに関する論理演算を行って、前記パス式による前記データベース検索におけるアクセス権を判定する論理演算部と

を備えることを特徴とする請求項１０に記載のデータベース検索システム。

【請求項１２】　前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出部と、

前記問い合わせ式から抽出された個々の前記パス式に対する前記論理演算部によるアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定部と

をさらに備えることを特徴とする請求項１１に記載のデータベース検索システム。

【請求項１３】　前記アクセス権解析器は、

前記データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理部と、

前記データベースに対する検索条件を記述したパス式によって前記パステーブル管理部に管理されている前記パステーブル中の所定のパスを選択し、前記アク

セス制御規則を記述した前記アクセス制御ポリシーを適用して、当該所定のパスに対する当該パス式によるデータベース検索におけるアクセス権を判定するアクセス権判定部と

を備えることを特徴とする請求項１０に記載のデータベース検索システム。

【請求項１４】　前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出部と、

前記問い合わせ式から抽出された個々の前記パス式に対する前記アクセス権判定部によるアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定部と

をさらに備えることを特徴とする請求項１３に記載のデータベース検索システム。

【請求項１５】　コンピュータを用いてＸＭＬ文書を格納したデータベースへのアクセス権を解析するアクセス権解析方法であって、

前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成し、アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成し、生成された当該検索オートマトンおよび当該アクセス制御オートマトンを所定の記憶手段に格納するステップと、

前記所定の記憶手段に格納された前記検索オートマトンおよび前記アクセス制御オートマトンに関する論理演算を行って、前記データベースに格納された前記ＸＭＬ文書を調べることなく、前記パス式によるデータベース検索におけるアクセス権を判定するステップと

を含むことを特徴とするアクセス権解析方法。

【請求項１６】　コンピュータを用いてＸＭＬ文書を格納したデータベースへのアクセス権を解析するアクセス権解析方法であって、

前記データベースに対する検索条件を記述したパス式によって、所定の記憶手段に格納され前記データベースに格納されているデータファイルのパスを記述したパステーブルから所定のパスを選択するステップと、

アクセス制御規則を記述したアクセス制御ポリシーを適用して、前記データベースに格納された前記データファイルを調べることなく、前記所定のパスに対す

る前記パス式によるデータベース検索におけるアクセス権を判定するステップとを含むことを特徴とするアクセス権解析方法。

【請求項１７】　コンピュータを制御して、構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析するプログラムであって、

前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成手段と、

アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成手段と、

生成された前記検索オートマトンおよび前記アクセス制御オートマトンに関する論理演算を行って、前記パス式によるデータベース検索におけるアクセス権を判定する論理演算手段として

前記コンピュータを機能させることを特徴とするプログラム。

【請求項１８】　前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出手段と、

前記問い合わせ式から抽出された個々の前記パス式に対するアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定手段として

前記コンピュータをさらに機能させることを特徴とする請求項１７に記載のプログラム。

【請求項１９】　コンピュータを制御して、構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析するプログラムであって、

前記データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理手段と、

前記データベースに対する検索条件を記述したパス式によって前記パステーブル管理手段に管理されている前記パステーブル中の所定のパスを選択し、アクセス制御規則を記述したアクセス制御ポリシーを適用して、当該所定のパスに対する当該パス式によるデータベース検索におけるアクセス権の有無を判定するアクセス権判定手段として

前記コンピュータを機能させることを特徴とするプログラム。

【請求項２０】　前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出手段と、

前記問い合わせ式から抽出された個々の前記パス式に対するアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定手段として

前記コンピュータをさらに機能させることを特徴とする請求項１９に記載のプログラム。

【請求項２１】　請求項１７乃至請求項２０のいずれかに記載のプログラムを、コンピュータが読み取り可能に記録した記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、アクセス制御に関し、特にＸＭＬ（Extensible Markup Language）で記述された文書ファイルを扱うデータベースのためのアクセス制御に関する。

【０００２】

【従来の技術】

ＸＭＬで記述された文書（ＸＭＬ文書）は、要素と属性からなる木構造にて表現することができる。この木構造についての条件をスキーマによって記述することができる。したがって、このスキーマを用いることにより、特定の条件を満たす木構造をもつＸＭＬ文書のみを許容するようなシステムを作ることができる（スキーマについては、例えば、非特許文献１参照）。

ＸＭＬ文書を扱うデータベース（ＸＭＬデータベース）では、文書を取り出すだけではなく、ＸＭＬ文書中の要素や属性を限定して選択的に取り出すことができる。例えば、段落を表す要素（＜ｐ＞．．．＜／ｐ＞で表現される要素）を全て取り出すといった処理が可能である。検索条件を記述する方法としては、ＸＰａｔｈというＷ３Ｃ（World Wide Web Consortium）勧告が広く用いられている。

【０００３】

また、ＸＭＬデータベースでは、単に要素や属性を検索するだけではなく、取

り出された要素や属性を用いて、新たなＸＭＬ文書を作り出すこともできる。そのような機能を備えた検索言語として、ＸＱｕｅｒｙという仕様がＷ３Ｃで設計されつつある。ＸＱｕｅｒｙは、要素や属性を取り出すための機構としてＸＰａｔｈを用いる。

【０００４】

さらにＸＭＬデータベースにおいて、ＸＭＬ文書を構成する要素や属性に対してアクセスを行う必要がある場合について、具体例として、ある企業の従業員の一覧表を表現したＸＭＬ文書を考える。各従業員については、その年収や社員番号が示されている。従業員、年収、社員番号は、それぞれｅｍｐｌｏｙｅｅ要素、ｓａｌａｒｙ要素、ｎｕｍｂｅｒ要素によって表現されるものとする。このようなＸＭＬ文書を格納したＸＭＬデータベースでは、ｎｕｍｂｅｒ要素についてのアクセスは特に制限する必要はないが、ｓａｌａｒｙ要素についてのアクセスは一部の人に限定しなければならないという場合がある。誰がどの要素・属性をアクセスできるかについての記述を、アクセス制御ポリシーと呼ぶ。ユーザは、アクセス制御ポリシーを記述して、ＸＭＬデータベースに与える。ＸＭＬデータベースは、ＸＭＬ文書に対するアクセス要求に対して、与えられたアクセス制御ポリシーを利用して、要素・属性へのアクセスを許可するのか拒否するのかを判定する。

【０００５】

アクセス制御ポリシーを表現するための言語にＸＡＣＬ（XML Access Control Language）がある（ＸＡＣＬについては、例えば、非特許文献２参照）。ＸＡＣＬの言語仕様には、所定のアクセス要求に対してアクセス許可・拒否を決めるアルゴリズムが記述されている。このアルゴリズムは、所定の一つのノードに対する判定を行う。したがって、ＸＰａｔｈもしくはＸＰａｔｈによる検索が複数のノードをアクセスする場合には、どのノードに対してもこのアルゴリズムを一度ずつ実行する。

【０００６】

【非特許文献１】

村田真著、「XML [I] - XML SchemaとRelax -」、電子情報通信学会

誌、２００１年、１２月、Ｖｏｌ．８４、Ｎｏ．１２、ｐ．８９０－８９４

【非特許文献２】

工藤道治（Ｍｉｃｈｉｈａｒｕ　Ｋｕｄｏ）、羽田知史（Ｓａｔｏｓｈｉ　Ｈａｄａ）著、“XML Document Security based on Provisional Authorization”、Proceedings of the 7th ACM Conference on Computer and Communications Security、２０００年、１１月、ｐ．８７－９６

【非特許文献３】

Ｊ．ホップクロフト・Ｊ．ウルマン著（野崎・高橋・町田・山崎訳）、「オートマトン　言語理論　計算論Ｉ、ＩＩ」、サイエンス社、１９８６年

【非特許文献４】

“XQuery 1.0: An XML Query Language”、[online]、２００２年４月３０日、W3C Working Draft、[平成１５年１月２９日検索]、インターネット＜ＵＲＬ：HYPERLINK "http://www.w3.org/TR/2002/WD-xquery-20020430/" http://www.w3.org/TR/xquery/＞

【０００７】

【発明が解決しようとする課題】

上述したように、ＸＭＬデータベースでは、アクセス要求に対して、与えられたアクセス制御ポリシーを利用して、要素・属性へのアクセスを許可するのか拒否するのかを判定する必要があるが、このＸＭＬデータベースの性能を向上させるには、アクセス権限の判定を高速に行うことが必要となる。

しかし、アクセス制御ポリシーを表現する言語であるＸＡＣＬの言語仕様に示されたアクセス権判定アルゴリズムは高速とは言えない。しかも上述のように、実際のアクセス権判定では、このアルゴリズムによる処理が多くのノードに対して繰り返されることとなるため、その実行性能は実用的とは言えない。

【０００８】

なお、上記従来の技術では、ＸＭＬデータベースの検索条件の記述方法としてＸＰａｔｈ及びＸＱｕｅｒｙを挙げたが、このＸＰａｔｈに類似するもの（パス式と呼ぶ）やＸＱｕｅｒｙに類似するもの（問い合わせ式と呼ぶ）は他にも存在する。しかし、これらによるＸＭＬデータベースは皆、

・パス式（path expression）（又はパス式を含む問い合わせ式）
・ＸＭＬ文書（XML document）
を与えられて
・ＸＭＬ文書中のノード（要素・属性・テキスト）
を検索する。そして、検索されたノードに対するアクセスが許されているかどうかを
・アクセス制御ポリシー（access control policy）
と照合して判定する。
といった手順で処理を実行する点は同様である。したがって、いずれの方法であっても、例えば検索された要素・属性が１０００個あれば、この判定も１０００回繰り返されるため、この処理に多大な時間を要することは、共通する問題点と言える。

【０００９】

そこで本発明は、ＸＭＬ等の構造化文書で記述されたデータファイルを扱うデータベースにおいて、データファイル自体やそのノードを調べることなくアクセス権解析（access rights analysis）を行うことを実現し、データベースの検索性能を向上させることを目的とする。

【００１０】

【課題を解決するための手段】

上記の目的を達成する本発明は、ＸＭＬのような構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析する、次のように構成された情報処理装置として実現される。この情報処理装置は、データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部と、アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部と、データベースに格納されているデータファイルの構造を示すスキーマからスキーマオートマトンを生成するスキーマオートマトン生成部と、生成された検索オートマトン、アクセス制御オートマトン及びスキーマオートマトンに関する論理演算を行って、かかるパス式によるデータベース検索におけるアクセス権を判定する論理演算部とを備

えることを特徴とする。

【００１１】

また、本発明の情報処理装置は、データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理部をさらに備え、このパステーブルから直接、もしくはスキーマオートマトンを生成して、論理演算部がアクセス権を判定する構成とすることができる。

さらに、データベースに対する検索方法を指定する問い合わせ式からアクセス権解析の対象となるパス式を抽出することができる。この場合、問い合わせ式から抽出された個々のパス式に対する論理演算部によるアクセス権の判定結果に基づいて、問い合わせ式によるデータベース検索におけるアクセス権を判定する。

【００１２】

さらにまた、上記の目的を達成する他の本発明は、コンピュータを用いてＸＭＬ文書を格納したデータベースへのアクセス権を解析する、次のようなアクセス権解析方法としても実現される。このアクセス権制御方法は、データベースに対する検索条件を記述したパス式から検索オートマトンを生成し、アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成し、生成された検索オートマトンおよびアクセス制御オートマトンを所定の記憶手段に格納するステップと、これら検索オートマトンおよびアクセス制御オートマトンに関する論理演算を行って、データベースに格納されたＸＭＬ文書を調べることなく、パス式によるデータベース検索におけるアクセス権を判定するステップとを含むことを特徴とする。

【００１３】

また、本発明の他のアクセス権解析方法は、データベースに対する検索条件を記述したパス式によって、所定の記憶手段に格納され、データベースに格納されているデータファイルのパスを記述したパステーブルから所定のパスを選択するステップと、アクセス制御規則を記述したアクセス制御ポリシーを適用して、データベースに格納されたデータファイルを調べることなく、所定のパスに対するパス式によるデータベース検索におけるアクセス権を判定するステップとを含むことを特徴とする。

【００１４】

さらに本発明は、コンピュータを制御して上述したアクセス権解析器あるいはデータベース検索システムの各機能を実現させるプログラム、またはコンピュータに上記アクセス権解析方法の各ステップに相当する処理を実行させるプログラムとしても実現される。このプログラムは、磁気ディスクや光ディスク、半導体メモリ、その他の記録媒体に格納して配布したり、ネットワークを介して配信したりすることにより、提供される。

【００１５】

【発明の実施の形態】

以下、添付図面に示す実施の形態に基づいて、この発明を詳細に説明する。

本発明は、ＸＭＬデータベースに対するアクセス権解析において、ＸＭＬ文書の構造を調べて行う実際の解析に先立って、スキーマもしくはパステーブルを用いることにより予備的なアクセス権解析を実行する。この予備的な解析において、検索条件を記述する所定のパス式によるＸＭＬデータベースへのアクセスが、常に許可される（always permitted）か、常に拒否される（always denied）か、またはアクセス可否が不確定（indeterminate）であるかを、ＸＭＬ文書の検索前に判定する。そして、アクセス可否が不確定と判定した場合にのみ、ＸＭＬ文書の構造を調べて行う通常のアクセス権解析を行う。

【００１６】

検索条件を記述したパス式とアクセス条件を示すアクセス制御ポリシーとの組合せによっては、ＸＭＬ文書の検索前の段階で、アクセス権の解析が可能な場合がある。解析可能な例をいくつか挙げる。

例１：アクセス制御ポリシーは、ユーザｇｕｅｓｔが要素ｐにアクセスすることを許しているものとする。ＸＰａｔｈ式（パス式）は／／ｐとする。これは、タグ名ｐを持つ要素すべてを取り出す。取り出された要素は、ユーザｇｕｅｓｔによるアクセスが可能なことがあらかじめ保障されている。どんな文書であっても、アクセス権のない要素を／／ｐが取り出すことはあり得ない。この場合、１つ１つの要素ｐに対してアクセス権判定をする必要はない。

例２：逆に、要素ｐへのユーザｇｕｅｓｔによるアクセスが禁止されているとする。このとき、ＸＰａｔｈ式／／ｐが探し出す要素へのアクセス権をユーザｇｕｅｓｔが持たないことは明らかである。この場合、／／ｐで要素を探し出すことすら必要ではない。単に、アクセス権がないとみなして処理すれば良い。

例３：アクセス制御ポリシーが、ｓｅｃ要素の子孫であるｐ要素へのアクセスを許しているとする。ＸＰａｔｈ式は／／ｓｅｃ／／ｐとする。これは、タグ名ｐを持つ要素であって、タグ名ｓｅｃを持つ要素の子孫であるもの全てを取り出す。この場合、取り出された要素に、ユーザｇｕｅｓｔがアクセス可能なことは明らかである。

例４：アクセス制御ポリシーは、例３と同様とする。そして、スキーマが存在し、このスキーマは、タグ名ｐを持つ要素を、タグ名ｓｅｃを持つ要素の子要素としてしか許さないものとする。この場合、ＸＰａｔｈ式／／ｐが取り出す要素に、ユーザｇｕｅｓｔによるアクセスが可能なことは明らかである。

例５：アクセス制御ポリシーが、ｓｅｃ要素およびその子孫へのアクセスを禁止しているものとする。ＸＰａｔｈ式は／／ｓｅｃ／／ｐとする。このＸＰａｔｈ式が探し出す要素へのアクセス権がないことは明らかである。

例６：アクセス制御ポリシーは、例５と同様とする。そして、スキーマが存在し、このスキーマは、タグ名ｐを持つ要素を、タグ名ｓｅｃを持つ要素の子要素としてしか許さないものとする。この場合も、ＸＰａｔｈ式／／ｐが探し出す要素へのアクセス権がないことは明らかである。

【００１７】

これらは、ＸＭＬ文書やそのノードを調べることなくアクセス権解析が可能な場合の一例に過ぎない。このような予備的なアクセス権解析によりアクセス判定

が可能な場合は、ＸＭＬ文書の検索前にこの判定を一度だけ行えば良く、検索されたＸＭＬ文書のノードごとにアクセス判定を繰り返す必要がないため、大幅な性能向上を望むことができる。

以下では、パス式（ＸＰａｔｈ）を用いたアクセス権解析（実施の形態１、２）、パス式を内包する問い合わせ式（ＸＱｕｅｒｙ）を用いたアクセス権解析（実施の形態３）、これらの実装例（実施の形態４、５）に分けて本発明を説明する。

【００１８】

〔実施の形態１〕

本実施の形態によるアクセス権解析では、アクセス制御ポリシー（access control policy）からアクセス制御オートマトン（access control automaton）Γを生成し、スキーマ（schema）からスキーマオートマトン（schema automaton）Ｓを生成する。そして、アクセス制御オートマトンΓとスキーマオートマトンＳとを用いて、所定のパス式ｑに対しアクセス可否を判定する。

オートマトン（automaton）は、いくつかのシンボルから構成された列を調べて、受理されるかどうかを判定する機構として広く知られている。オートマトンの定義やオートマトンに対する論理演算については、例えば非特許文献３に、詳細に記述されている。また、シンボル列の集合を記述する方法として、正規表現（regular expression）がある。正規表現は、オートマトンに変換することができ、その方法も、非特許文献３に記述されている。

【００１９】

ここで、パスとは、

・ＸＭＬ文書のルート要素からある要素（木構造の葉でなくてもよい）に至るまでのタグ名を並べたもの

例：／ａ／ｂ／ｃ（ここで、ａ、ｂ、ｃ、はタグ名）

・ＸＭＬ文書のルート要素からある要素（木構造の葉でなくてもよい）に至るまでのタグ名を並べ、この要素のある属性の属性名を追加したもの

例：／ａ／ｂ／ｃ／＠ｄ（ここで、ｄは属性名）

・ＸＭＬ文書のルート要素からある要素（木構造の葉でなくてもよい）に至るま

でのタグ名を並べ、この要素のテキストを参照するという情報を追加したもの
例：／ａ／ｂ／ｃ／ｔｅｘｔ（）
のいずれかとする。

パス式（path expression）とは、パスについての条件を表す式である。例えば、「ルート要素ｆｏｏから出発して要素ｐに至る」は、パス式である。パス式の１つであるＸＰａｔｈでは、この条件式を／ｆｏｏ／／ｐと表現する。本実施の形態では、このＸＰａｔｈをパス式として用いる。

【００２０】

アクセス制御ポリシーは、いくつかのアクセス制御規則（access control rule）からなる。各規則は、パス式を用いてアクセスの可否を記述する。あるノードのパスが、このパス式に合致するとき、アクセスは許可または拒否される。上述のように、本実施の形態では、ＸＭＬ文書中のノードに対するアクセス権を判定するために、アクセス制御オートマトンを導入する。アクセス制御オートマトンとは、ＸＭＬ文書のルート要素からあるノードまでのパスが受理されるなら、またそのときに限ってアクセス権があるようなオートマトンである。

さらに本実施の形態では、ＸＭＬ文書中のノードをＸＰａｔｈによって取り出すために、検索オートマトン（query automaton）を導入する。検索オートマトンとは、ＸＭＬ文書のルート要素からあるノードまでのパスが受理されるなら、またそのときに限って取り出すようなオートマトンである。検索オートマトンはＸＰａｔｈ式から生成する。

【００２１】

また、本実施の形態において、スキーマとは、

非終端記号ｘ ::＝ 正規表現ｒ

という形をした規則がいくつか並んだものとする。正規表現ｒを構成するシンボルは、タグ名と非終端記号ｘの対または属性名と非終端記号ｘの対である。いくつかの非終端記号ｘは、開始記号として指定されている。スキーマを表現するための言語として、ＤＴＤ（Document Type Definition）、W3C XML Schema、RELAX（REgular LAnguage description for XML） NGなどが知られている。これらのスキーマ言語で書かれたスキーマを、上記の形のスキーマに変換することは容易

である。したがって本実施の形態は、これらのスキーマ言語で書かれたスキーマにも適用されるものである。

上述のように、本実施の形態では、アクセス制御オートマトンおよび検索オートマトンとスキーマを照合するため、スキーマオートマトンを導入する。スキーマオートマトンとは、スキーマに照らして妥当な文書のルートノードから任意のノードからまでのパスを全て受理し、それ以外のパスは受理しないオートマトンである。

【００２２】

図１は、本実施の形態によるアクセス権解析器を実現するのに好適なコンピュータ装置のハードウェア構成の例を模式的に示した図である。

図１に示すコンピュータ装置は、演算手段であるＣＰＵ（Central Processing Unit：中央処理装置）１０１と、Ｍ／Ｂ（マザーボード）チップセット１０２及びＣＰＵバスを介してＣＰＵ１０１に接続されたメインメモリ１０３と、同じくＭ／Ｂチップセット１０２及びＡＧＰ（Accelerated Graphics Port）を介してＣＰＵ１０１に接続されたビデオカード１０４と、ＰＣＩ（Peripheral Component Interconnect）バスを介してＭ／Ｂチップセット１０２に接続されたハードディスク１０５、ネットワークインターフェイス１０６及びＵＳＢポート１０７と、さらにこのＰＣＩバスからブリッジ回路１０８及びＩＳＡ（Industry Standard Architecture）バスなどの低速なバスを介してＭ／Ｂチップセット１０２に接続されたフロッピーディスクドライブ１０９及びキーボード／マウス１１０とを備える。

なお、図１は本実施の形態を実現するコンピュータ装置のハードウェア構成を例示するに過ぎず、本実施の形態を適用可能であれば、他の種々の構成を取ることができる。例えば、ビデオカード１０４を設ける代わりに、ビデオメモリのみを搭載し、ＣＰＵ１０１にてイメージデータを処理する構成としても良いし、ＡＴＡ（AT Attachment）などのインターフェイスを介してＣＤ－ＲＯＭ（Compact Disc Read Only Memory）やＤＶＤ－ＲＯＭ（Digital Versatile Disc Read Only Memory）のドライブを設けても良い。

【００２３】

図２は、本実施の形態によるアクセス権解析器の機能構成を示すブロック図である。

図２に示すように、本実施の形態によるアクセス権解析器２００は、パス式から検索オートマトンを構築する検索オートマトン生成部２１０と、アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部２２０と、スキーマからスキーマオートマトンを生成するスキーマオートマトン生成部２３０と、これらのオートマトンを用いてアクセス権の判定を行う論理演算部２４０とを備える。そして、ＸＭＬデータベースの前段でＸＭＬ文書の検索前に予備的なアクセス権解析を実行する。

【００２４】

上述した各構成要素は、例えば、図１に示したメインメモリ１０３に展開されたプログラムにてＣＰＵ１０１を制御することにより実現される仮想的なソフトウェアブロックである。ＣＰＵ１０１を制御してこれらの機能を実現させるプログラムは、磁気ディスクや光ディスク、半導体メモリ、その他の記憶媒体に格納して配布したり、ネットワークを介して配信したりすることにより提供される。本実施の形態では、図１に示したネットワークインターフェイス１０６やフロッピーディスクドライブ１０９、図示しないＣＤ－ＲＯＭドライブなどを介して当該プログラムを入力し、ハードディスク１０５に格納する。そして、ハードディスク１０５に格納されたプログラムをメインメモリ１０３に読み込んで展開し、ＣＰＵ１０１にて実行することにより、これらの機能を実現する。

【００２５】

上記の構成において、検索オートマトン生成部２１０は、検索条件を指定するパス式を入力して検索オートマトンを生成する。具体的には、解析すべきＸＰａｔｈ式が与えられると、これを正規パス式とみなし、さらに検索オートマトンに変換したものを検索オートマトンｑで表現する。生成された検索オートマトンは、例えば図１に示したメインメモリ１０３の作業領域に保持され、論理演算部２４０にて使用される。

ここで、本実施形態では、ＸＰａｔｈを正規パス式（regular path expression＝正規表現によって所定のノードに至るパスを表した制約式）とみなす手法を

用いる。すなわち、

・「／」は、正規表現εとみなす。ここで、εは空文字をあらわす。

・「／／ｔｉｔｌｅ」は、正規表現Σ^*．ｔｉｔｌｅとみなす。ここで、Σ^*は、任意の要素名の任意の回数の繰り返しをあらわす。また「．」は連結を示し、Σ^*．ｔｉｔｌｅとは、任意の要素名の繰り返しの後にｔｉｔｌｅがくるようなＸＭＬ文書のパスを意味する。

・「／ｂｉｂ／ｂｏｏｋ／ｔｉｔｌｅ」は、正規表現ｂｉｂ．ｂｏｏｋ．ｔｉｔｌｅとみなす。これは、ｂｉｂ、ｂｏｏｋ、ｔｉｔｌｅをこの順で通るようなパスを意味する。

所定のノード（要素，属性，テキスト）ｎに至るパスがパス正規表現ｒに属するならば、ノードｎはパス正規表現ｒに選択されるという。

【００２６】

図３は検索対象となるＸＭＬ文書の例を示す図である。

図３のＸＭＬ文書に対する検索を考えると、検索したいＸＰａｔｈ式が全てのａｕｔｈｏｒ要素、すなわち／／ａｕｔｈｏｒであるならば検索オートマトンｑは、図４に示すオートマトンになる。また、ＸＰａｔｈ式がｂｉｂ要素の下のｂｏｏｋ要素の全ての子供、すなわち／ｂｉｂ／ｂｏｏｋ／＊であるならば、検索オートマトンｑは、図５に示すオートマトンになる。

【００２７】

アクセス制御オートマトン生成部２２０は、アクセス制御のために与えられたアクセス制御ポリシーからアクセス制御オートマトンを生成する。生成されたアクセス制御オートマトンは、例えば図１に示したメインメモリ１０３の作業領域に保持され、論理演算部２４０にて使用される。

アクセス制御ポリシーをＸＡＣＬ２（ＸＡＣＬのアクセス制御の伝播を下記のように定義したもの）で記述する場合、そのアクセス制御規則は、サブジェクト（検索の主体）、ＸＰａｔｈ、アクション（アクセスの内容と可否の情報）の組で表される。

ＸＡＣＬ２の各アクセス制御規則が、次のように記述されているとする。

（Ａｄｍｉｎ，／，＋ｒ）
（Ｇｕｅｓｔ，／ｂｉｂ／ｂｏｏｋ，＋ｒ）
（Ｇｕｅｓｔ，／／ｐｒｉｃｅ，－ｒ）

一方、ＸＡＣＬ２におけるリード・ポリシーの解釈には、次のような原則がある。

・読み込み許可＋ｒ（grant read +r）は下に伝播する。
・読み込み禁止－ｒ（deny read -r）は下に伝播する。
・－ｒは＋ｒに優先し、またデフォルトは－ｒである。

すなわち、Ｇｕｅｓｔが図３のＸＭＬ文書をみるとき、／ｂｉｂ／ｂｏｏｋの＋ｒが下に伝播し、また／／ｐｒｉｃｅの－ｒが優先であるために、Ｇｕｅｓｔにみえるのはｐｒｉｃｅ要素を除いた全てのノードである。

【００２８】

本実施の形態では、サブジェクトが予め与えられているとする。サブジェクトが可変である一般の場合は実施の形態４、５で検討する。

アクセス制御オートマトン生成部２２０は、上述したようなポリシー記述（＋ｒ、－ｒ）とサブジェクト（Ａｄｍｉｎ、Ｇｕｅｓｔ）が与えられたとき、次のオートマトンR^+、R^-を計算する。

・オートマトンR^+は、そのサブジェクトに対するアクション＋ｒの許されたＸＰａｔｈに対応するパス正規表現に対応するオートマトン全ての和
・オートマトンR^-は、そのサブジェクトに対するアクション－ｒの許されたＸＰａｔｈに対応するパス正規表現に対応するオートマトン全ての和

正規表現からオートマトンへの変換およびオートマトンに対する集合演算のアルゴリズムは周知である（例えば、非特許文献３参照）。さらに以下のようなアクセス制御オートマトンΓを計算する。

$$\Gamma = R^+ . \Sigma^* - R^- . \Sigma^*$$

すなわち、アクセス制御オートマトンΓは、オートマトンR^+の後に任意のパ

スΣ*を連結した正規集合から、オートマトンR⁻の後に任意のパスΣ*を連結した正規集合を取り除いた差分の正規集合を表現するオートマトンである。

【００２９】

例を挙げると、上記のゲストに対するオートマトンR⁺は、ε｜ｂｉｂ．ｂｏｏｋという正規表現で表すことができ、オートマトンR⁻は、Σ*．ｐｒｉｃｅで表すことができるので、アクセス制御オートマトンΓは、（ε｜ｂｉｂ．ｂｏｏｋ）．（Σ－｛ｐｒｉｃｅ｝）*という正規表現で示すことができる。

図６は、かかるアクセス制御オートマトンΓを説明する図である。

図６に示すアクセス制御オートマトンΓにおいて、ＸＭＬ文書のインスタンスに対するアクセス判定（access decision）は、次のようにする。

ＸＭＬ文書において、読み込みアクセスが許可されたノード（要素、属性、テキスト等）の集合ＧＲＡＮＴ、読み込みアクセス権がないノードの集合ＤＥＮＹを次のように求めことができる。

ｎ∈ＧＲＡＮＴ⇔ノードｎは、アクセス制御オートマトンΓで選択される。

ＤＥＮＹ＝「与えられた文書中の全てのノードの集合」－ＧＲＡＮＴ

ただし、所定のノード（要素、属性、テキスト）ｎに至るパスがオートマトンΓに受理されるならば、ノードｎはアクセス制御オートマトンΓに選択されると言う。

【００３０】

スキーマオートマトン生成部２３０は、スキーマの記述からスキーマオートマトンを生成する。生成されたスキーマオートマトンは、例えば図１に示したメインメモリ１０３の作業領域に保持され、論理演算部２４０にて使用される。ＸＭＬ文書のためのスキーマとしては、ＤＴＤ、XML Schema、RELAX NG等がよく用いられている。本実施の形態では、ＤＴＤを用いるものとして説明するが、他の種類のスキーマでも概ね同様である。

次のようなＤＴＤを考える。

```
<!ELEMENT bib (book*)>
<!ELEMENT book (title, author*, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

【００３１】

本実施の形態では、このＤＴＤで許されている任意のパスに対応するパス正規表現に対応するオートマトンＳを考える。このＤＴＤで許されるどのＸＭＬ文書のどのノードに対するルートからのパスも、このオートマトンＳによって受理され、それ以外のパスはオートマトンＳによって受理されない。このオートマトンＳがスキーマオートマトンである。

ＤＴＤを与えられてスキーマオートマトンＳを求めるアルゴリズムを以下に示す。

まず、スキーマオートマトンＳの状態遷移を求める。各要素名ｎに対する状態をｑｎとする。いま、<!ELEMENT n (…)>のような要素宣言（element declaration）があった場合、（…）の中に要素名ｍが出現するとき、要素名ｎと状態ｑｎとを与えられて、ｑｍに遷移する遷移をスキーマオートマトンＳに追加する。これを全ての要素宣言に対して行えば、スキーマオートマトンＳの状態遷移（state transition）が求まる。スキーマオートマトンＳの初期状態（intial state）は、ルートの要素名に対応する状態である。また、スキーマオートマトンＳの終了状態集合（final state set）は、全ての状態である。

例えば上記のＤＴＤからは、bib.(ε|book.(ε|title|author|price))という正規表現に対応するスキーマオートマトンＳが得られる。

【００３２】

論理演算部２４０は、上記の検索オートマトン生成部２１０、アクセス制御オートマトン生成部２２０及びスキーマオートマトン生成部２３０により生成された各オートマトンを用いて、与えられたパス式によるＸＭＬ文書の所望のノードへのアクセス可否を判定する。ここで判定しようとする内容は、次の２つの性質

である。

・常に拒否（always denied）：検索オートマトンｑで検索される全てのノードへのアクセスが拒否されるか否か。

・常に許可（always permitted）：検索オートマトンｑで検索される全てのノードへのアクセスが許可されるか否か。

「常に許可」の判定にも「常に拒否」の判定にも失敗した場合、論理演算部２４０は、「不確定（indeterminate）」という結果を出力する。

【００３３】

「常に許可」と「常に拒否」を判定するためには、次のことを調べればよい（ただし、ｑは検索オートマトン、Γはアクセス制御オートマトン、Ｓはスキーマオートマトン）。

「常に拒否」　ｑがΓ∩Ｓとｄｉｓｊｏｉｎｔ

「常に許可」　ｑ∩ＳがΓに含まれる

ここで、Ｘ∩Ｙとは、ＸとＹの積集合を示す。ＸとＹとがｄｉｓｊｏｉｎｔであるとは、Ｘ∩Ｙが空集合であることを意味する。含まれるとは、集合の包含⊆を示す。これらのオートマトンに対する演算は、全て、例えば非特許文献３に開示された既存の手法を用いて計算することができる。

【００３４】

上述したＸＰａｔｈ式／／ａｕｔｈｏｒに対応する検索オートマトンｑとＤＴＤにおけるスキーマオートマトンＳとの共通部分は、bib.book.authorに対応するオートマトンである。これは、上述したアクセス制御ポリシーにおけるＧｕｅｓｔに対するアクセス制御オートマトンΓに含まれている。よって、ＸＰａｔｈ式／／ａｕｔｈｏｒは「常に許可」と判定される。

一方、上述したＸＰａｔｈ式／ｂｉｂ／ｂｏｏｋ／＊に対応する検索オートマトンｑについてスキーマオートマトンＳ、アクセス制御オートマトンΓを適用すると、次のようになる。

まず、ｑ∩Ｓは、bib.book.(author|title|price)に対応するオートマトンで

ある。アクセス制御オートマトンΓは、このオートマトンを含まない。次に、Ｓ∩Γはε|bib.(ε|book.(ε|title|author))に対応する、このオートマトンは、bib.book.Σのオートマトンと共通部分を持つ。よって、ＸＰａｔｈ式／ｂｉｂ／ｂｏｏｋ／＊は、「常に許可」でも「常に拒否」でもなく「不確定」である。

【００３５】

ここで補足として、上述した判定規則がどうして正しいのかの説明をする。

＜「常に拒否」の判定規則＞

「ｑがΓ∩Ｓとｄｉｓｊｏｉｎｔ」ならば「ｑで検索されるノード集合⊆ＤＥＮＹ」

集合ＧＲＡＮＴについて上述した次の性質を使う。

①　ｎ∈ＧＲＡＮＴ⇔ノードｎはΓで検索される。

いま、ドキュメントの全てのノードに対するパスがスキーマオートマトンＳによって受理されるとする。すると、

②　ノードｎがΓで検索される⇒ノードｎはΓ∩Ｓによって検索される。

①と②より、ｎ∈ＧＲＡＮＴならば、ノードｎはΓ∩Ｓによって検索されるということがわかる。したがって、「ｑがΓ∩Ｓとｄｉｓｊｏｉｎｔ」であるならば、ｑで検索されるノードはΓ∩Ｓによっては検索されないので、ＧＲＡＮＴではない。

【００３６】

したがって、上記の「常に拒否」の規則による判定が成功すれば、検索オートマトンｑによる検索結果は必ず集合ＤＥＮＹに含まれることがわかる。

図７は、「常に拒否」の判定規則を説明する図である。

図７において、ノードｎ１は検索オートマトンｑで検索されるノード、ノードｎ２、ｎ３はアクセス制御オートマトンΓで検索されるノードである。すなわち、検索オートマトンｑで検索されるノードは、アクセス制御オートマトンΓで検

索されるノードに含まれないために、必ずＤＥＮＹである。

【００３７】

＜「常に許可」の判定規則＞

「ｑ∩ＳがΓに含まれる」ならば「ｑで検索されるノード集合⊆ＧＲＡＮＴ」

もし、ｑ∩ＳがΓに含まれるならば、ｎ∈ｑ∩Ｓであるノードｎはアクセス制御オートマトンΓに検索されることになる。よって明らかにＧＲＡＮＴである。

したがって、上記の「常に許可」の規則による判定が成功すれば、検索オートマトンｑによる検索結果は必ず集合ＧＲＡＮＴに含まれることがわかる。

ここで、上記の「常に許可」の判定は、アクセス制御オートマトンΓから上方向に伝播した部分に対する検索が「常に許可」であることを利用することができない。ただし、これを利用する手法を採ることも可能である。

【００３８】

このように、本実施の形態によれば、ＸＭＬデータベース中のＸＭＬ文書に対して、データベース検索にかかるパス式のアクセス権が、少なくとも、「常に許可」、「常に拒否」、「不確定」のいずれに該当するかということが、当該ＸＭＬデータベースにおいて実際にＸＭＬ文書の構造を調べて行うアクセス権解析（ノード単位のアクセス権チェック）に先立って判断される。

「常に許可」である場合には、当該パス式を用いたデータベース検索では、ＸＭＬデータベースで、ＸＭＬ文書を調べて行う通常のアクセス権解析を行う必要はない。

「常に拒否」である場合は、当該パス式を用いたデータベース検索自体が無効であることが事前にわかる。

したがって、本実施の形態による判定結果が「不確定」である場合にのみ、ＸＭＬ文書を調べて行う通常のアクセス権解析をＸＭＬデータベースにおいて実行すれば良く、データベース検索の処理全体における実行効率を向上させることができる。

【００３９】

以上、本実施の形態では、アクセス制御オートマトンΓとスキーマオートマトンＳとを用いて、所定のパス式ｑに対しアクセス可否を判定するアクセス権解析

について説明した。ところで、ＸＭＬ文書には、ＤＴＤやその他のスキーマが指定されていないものも存在する。そのような場合、スキーマオートマトンＳは、Σ^*（任意の要素名の任意の回数の繰り返し）に対応するオートマトンであるとみなすことにより、上述した本実施の形態のアクセス権解析を適用することができる。

【００４０】

〔実施の形態２〕

上記の実施の形態１では、アクセス制御ポリシーからアクセス制御オートマトンΓを生成し、スキーマからスキーマオートマトンＳを生成して、このアクセス制御オートマトンΓとスキーマオートマトンＳとを用いて、所定のパス式ｑに対しアクセス可否を判定した。

これに対し、実施の形態２では、ＤＴＤなどのスキーマの記述の代わりにパステーブル（path table）を用いてアクセス権解析を行う。

【００４１】

ここで、パステーブルとは、ＸＭＬデータベースに格納されているＸＭＬ文書に存在する全てのノード（要素、属性、テキスト等）へのルートからのパスを記憶するデータ構造である。言い換えれば、ＸＭＬデータベース中のＸＭＬ文書における全てのパスを記述した一覧表である。ただし、本実施の形態では、ノードとして要素のみを扱う。

例えば、次のＸＭＬ文書に対するパステーブルを構築する場合を考える。

```
<ｆｏｏ　ａ０＝””>
　<ｂａｒ　ａ-１＝””>
　　<ｈｏｇｅ　ａ２＝””>テキスト</ｈｏｇｅ>
　</ｂａｒ>
　<ｂａｒ　ａ３＝””/>
</ｆｏｏ>
```

このＸＭＬ文書に含まれる全てのパスを列挙する（ただし、空のテキストを末

端とするパスは省略する）と、次のようになる。

／ｆｏｏ
／ｆｏｏ／@ａ０
／ｆｏｏ／ｂａｒ
／ｆｏｏ／ｂａｒ／@ａ１
／ｆｏｏ／ｂａｒ／@ａ３
／ｆｏｏ／ｂａｒ／ｈｏｇｅ
／ｆｏｏ／ｂａｒ／ｈｏｇｅ／ｔｅｘｔ（）
／ｆｏｏ／ｂａｒ／ｈｏｇｅ／@ａ２

これらをレコードとしてパステーブルが得られる。なお、先頭が@である名前は、属性を表す。

【００４２】

本実施の形態によるアクセス権解析器は、上述した実施の形態１と同様に、図１に示したコンピュータ装置等で実現される。

図８は、本実施の形態によるアクセス権解析器の機能構成を示すブロック図である。

図８に示すように、本実施の形態によるアクセス権解析器３００は、パス式から検索オートマトンを生成する検索オートマトン生成部２１０と、アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部２２０と、検索対象であるＸＭＬ文書のパステーブル３３１を管理するパステーブル管理部３３０と、これらのオートマトンを用いてアクセス権の判定を行うアクセス権判定部としての論理演算部３４０とを備える。

これらの構成要素のうち、検索オートマトン生成部２１０及びアクセス制御オートマトン生成部２２０は、実施の形態１における検索オートマトン生成部２１０及びアクセス制御オートマトン生成部２２０と同様であるので、同一の符号を付して説明を省略する。

【００４３】

パステーブル管理部３３０は、例えば、図１に示したプログラム制御されたＣＰＵ１０１及びメインメモリ１０３やハードディスク１０５等の記憶手段にて実現され、ＸＭＬデータベースに格納されているＸＭＬ文書に関して作成された、上述したパステーブル３３１を格納して管理する。パステーブル３３１は、公知の解析手段を用いてＸＭＬデータベース中のＸＭＬ文書を解析して予め作成しておくが、ＸＭＬデータベースの内容の変更に伴って適宜更新される。

図９は、図３に示したＸＭＬ文書に対して作成されるパステーブル３３１の例を示す図である。

図９のパステーブル３３１には、図３のＸＭＬ文書における全てのパスが列挙されている。

【００４４】

論理演算部３４０は、検索オートマトン生成部２１０及びアクセス制御オートマトン生成部２２０により生成されたオートマトンと、パステーブル管理部３３０に格納されているパステーブル３３１とを用いて、与えられたパス式によるＸＭＬ文書の所望のノードへのアクセス可否を判定する。ここで判定しようとする内容は、次の２つの性質である。

・常に拒否（always denied）：検索オートマトンｑで検索される全てのノードへのアクセスが拒否されるか否か。

・常に許可（always permitted）：検索オートマトンｑで検索される全てのノードへのアクセスが許可されるか否か。

「常に許可」の判定にも「常に拒否」の判定にも失敗した場合、論理演算部３４０は、「不確定（indeterminate）」という結果を出力する。

【００４５】

ＸＭＬデータベースに対してパステーブル３３１が存在する場合、スキーマオートマトンを生成することなく、このパステーブル３３１のみを参照してアクセス権解析を行うことができる。例えば、／ｂｉｂというＸＰａｔｈによる検索をする場合を考える。図９のパステーブル３３１が与えられており、アクセス制御ポリシーがＸＡＣＬ２にて次のように記述され（なお、＋ｒは読み込み許可、－ｒは読み込み禁止）、

（Ａｄｍｉｎ，／，＋ｒ）
（Ｇｕｅｓｔ，／ｂｉｂ／ｂｏｏｋ，＋ｒ）
（Ｇｕｅｓｔ，／／ｐｒｉｃｅ，－ｒ）

かつサブジェクトがＧｕｅｓｔであったものとする。

【００４６】

スキーマオートマトンを用いず、パステーブル３３１のみを用いてアクセス権解析を行うには、例えば、パステーブル３３１の各エントリに対して、アクセス制御においてどう判定されるかを示す情報（以下、判定情報）を予め登録しておき、この情報に従って判定を行うという手法を採ることができる。

図１０は、図９のパステーブル３３１にアクセス制御の判定情報を登録した様子を示す図である。

図１０に示すパステーブル３３１において、アクセス制御オートマトンΓで選択されるエントリは、／ｂｉｂ／ｂｏｏｋ、／ｂｉｂ／ｂｏｏｋ／ｔｉｔｌｅ、／ｂｉｂ／ｂｏｏｋ／ａｕｔｈｏｒの３つである。いま、検索オートマトン／ｂｉｂに対して、これで検索される全てのエントリ（＝ｑ∩Ｓ）に読み込み許可を示す＋ｒが付加されていたならば、判定結果は「常に許可」である。

反対に、検索オートマトンで検索される全てのパステーブルエントリに読み込み禁止を示すーｒが付加されていたならば、「ｑがΓ∩Ｓとｄｉｓｊｏｉｎｔ」であるという判定が成功したことになるため、判定結果は「常に拒否」である。

検索オートマトンで検索される全てのパステーブルエントリに、読み込み許可を示す＋ｒと読み込み禁止を示すーｒとが混在しているならば、判定結果は「不確定」となる。

【００４７】

さて、上記のアクセス権解析の手法では、スキーマオートマトンを生成せずに、パステーブル３３１の情報のみからアクセスの可否を判定した。これに対し、パステーブル３３１にかかる情報を登録しておくのではなく、パステーブル３３１のエントリからスキーマオートマトンを生成してアクセス権解析を行うことも

可能である。

すなわち、ＸＭＬデータベース中のＸＭＬ文書に対し、予めその内容から図９に示したようなパステーブル３３１が作られていれば、全てのＸＭＬ文書は、このパステーブル３３１に存在するパスに制約されるので、このパステーブル３３１をスキーマオートマトンとして用いることができる。

【００４８】

この場合、図８に示した本実施の形態のアクセス権解析器３００に、さらにスキーマオートマトン生成部が設けられ、パステーブル管理部３３０からパステーブル３３１を読み込んでスキーマオートマトンを生成し、論理演算部３４０に送ることとなる。

パステーブル３３１からスキーマオートマトンを生成するためには、パステーブル３３１の各パスにそれぞれ対応するオートマトンを作り、その和を求めるという手法を採ることができる。すなわち、／ｂｉｂ／ｂｏｏｋ／ｔｉｔｌｅであれば、ｂｉｂ．ｂｏｏｋ．ｔｉｔｌｅという正規表現に対応するオートマトンを作り、各オートマトンの和は既知の手法で求めれば良い。

【００４９】

このように、本実施の形態によれば、実施の形態１の場合と同様に、ＸＭＬデータベース中のＸＭＬ文書に対して、データベース検索にかかるパス式のアクセス権が、少なくとも、「常に許可」、「常に拒否」、「不確定」のいずれに該当するかということが、当該ＸＭＬデータベースにおいて実際にＸＭＬ文書の構造を調べて行うアクセス権解析（ノード単位のアクセス権チェック）に先立って判断される。

したがって、本実施の形態による判定結果が「不確定」である場合にのみ、ＸＭＬ文書を調べて行う通常のアクセス権解析をＸＭＬデータベースにおいて実行すれば良く、データベース検索の処理全体における実行効率を向上させることができる。

【００５０】

〔実施の形態３〕

上記の実施の形態１、２では、検索条件の記述としてＸＰａｔｈ等のパス式を

用いたアクセス権解析について説明したが、実施の形態３では、パス式を内包する問い合わせ式を用いたアクセス権解析を説明する。

ここで、問い合わせ式（query expression）とは、ＸＭＬデータベースへの検索を指定する式である。パス式によってノードを見つけ、それに対して何らかの操作（要素をいくつか追加して新たなＸＭＬ文書を構築する操作など）を行う。Ｗ３ＣのＸＱｕｅｒｙは、問い合わせ式である。ＸＱｕｅｒｙは、パス式の表現にＸＰａｔｈを用いる。本実施の形態では、問い合わせ式としてＸＱｕｅｒｙを用いた例について説明する。

【００５１】

本実施の形態によるアクセス権解析器は、上述した実施の形態１と同様に、図１に示したコンピュータ装置等で実現される。

図１１は、本実施の形態によるアクセス権解析器の機能構成を示すブロック図である。

図１１に示すように、本実施の形態によるアクセス権解析器４００は、パス式から検索オートマトンを構築する検索オートマトン生成部２１０と、アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部２２０と、スキーマからスキーマオートマトンを生成するスキーマオートマトン生成部２３０と、これらのオートマトンを用いて個々のパス式に対するアクセス権の判定を行う論理演算部２４０と、問い合わせ式からパス式を抽出するパス式抽出部４５０と、論理演算部２４０による判定結果に基づいて問い合わせ式自体のアクセス権の判定を行う問い合わせ式書き換え・判定部４６０とを備える。

これらの構成要素のうち、検索オートマトン生成部２１０、アクセス制御オートマトン生成部２２０、スキーマオートマトン生成部２３０及び論理演算部２４０は、実施の形態１における対応する構成要素と同様であるので、同一の符号を付して説明を省略する。

【００５２】

パス式抽出部４５０は、ＸＱｕｅｒｙ等の問い合わせ式を解析してパス式を抽出し、得られたパス式を検索オートマトン生成部２１０へ送る。パス式抽出部４

５０としては、例えば、図１に示したプログラム制御されたＣＰＵ１０１にて実現された汎用の構文解析器や、ＸＱｕｅｒｙ専用の構文解析器を用いることができる。

図１２は、問い合わせ式（ＸＱｕｅｒｙ）の例を示す図である。

図１２の問い合わせ式の場合、各命令は以下のように解釈され実行される。この手法は既存の公知技術である（例えば、非特許文献４参照）。

・１行目、８－９行目は、検索結果の全体を＜ｂｉｂ＞…＜／ｂｉｂ＞で囲む動作。

・２行目のｆｏｒ文は、ｈｔｔｐ：／／ｗｗｗ．ｂｎ．ｃｏｍで指定されたＸＭＬ文書に対して、／ｂｉｂ／ｂｏｏｋで選択される各ノードについて３－７行目の動作を繰り返すという動作。

・３行目のｗｈｅｒｅ文は、ｆｏｒ文で選択された各ノードを＄ｂとしたときに、＄ｂ／ｐｕｂｌｉｓｈｅｒのテキスト値が“Ａｄｄｉｓｏｎ－Ｗｅｓｌｅｙ”であり、＄ｂ／＠ｙｅａｒの値が１９９１以上である場合に、４－８行の動作を行うことを示す。

・４－７行目は、ｆｏｒ文で選択された各ノード＄ｂに対して、その＠ｙｅａｒ値と、ｔｉｔｌｅだけをコピーしたｂｏｏｋ要素を作る動作を示す。

以上の動作における／ｂｉｂ／ｂｏｏｋ、＄ｂ／ｐｕｂｌｉｓｈｅｒなどがパス式（ＸＰａｔｈ）である。

【００５３】

図１３は、図１２の問い合わせ式から抽出されるパス式の一覧を示す図表である。

図１３において、２行目のｄｏｃｕｍｅｎｔ（…）／ｂｉｂ／ｂｏｏｋと、６行目のｄｏｃｕｍｅｎｔ（…）／ｂｉｂ／ｂｏｏｋ／ｔｉｔｌｅとでは、ＸＰａｔｈの評価のされ方が異なることに注意する。前者では、ｂｏｏｋ要素の子孫の内容は問われず、各ｂｏｏｋ要素自体をｆｏｒ文で数えあげられれば良いが、６行目のｔｉｔｌｅ要素は検索の結果であるために、子孫ノードを含めた全ての内容が問われる。すなわち、／ｂｉｂ／ｂｏｏｋ／ｐｒｉｃｅへのアクセスが拒否されていたとしても、ｆｏｒ文の実行には何の影響もないが、／ｂｉｂ／ｂｏｏ

ｋ／ｔｉｔｌｅ／ｓｕｂｔｉｔｌｅへのアクセスが禁止されていた場合、ｒｅｔｕｒｎ文の実行のためにアクセス制御が必要になる。

このようなＸＰａｔｈの評価のされ方の違いを、図１３では、グループＡとグループＢという分類によって示している。ｆｏｒ文に対応するのはグループＡであり、ｒｅｔｕｒｎ文のように子孫ノードの内容を見る場合にはグループＢに分類される。また＄ｂ／ｐｕｂｌｉｓｈｅｒ＝“Ａｄｄｉｓｏｎ　Ｗｅｓｌｅｙ”のようにあるＸＰａｔｈ式の結果のテキスト値が問題となっている場合も、グループＢに分類されることになる。

【００５４】

各グループＡ、Ｂに充当するＸＰａｔｈ式の分類は以下の通りである。

グループＡ：子孫を考慮しなくて良いＸＰａｔｈ式。

Ｅｘ１：Location designator、ｆｏｒに現れる要素のＸＰａｔｈ式。

Ｅｘ２：Attribute、様々なところに現れるだろう属性のＸＰａｔｈ式。

Ｅｘ３：Location designator、様々なところに現れるだろう要素のＸＰａｔｈ式（／ｔｅｓｔ［．ｆｏｌｌｏｗｓ　ｘｍｌ］における／ｔｅｓｔ／ｘｍｌや／ｔｅｓｔ／．）。

Ｅｘ４：Location designator、様々なところに現れるだろう要素のＸＰａｔｈ式（ｎａｍｅｓｐａｃｅ－ｕｒｉ（．／ｔｅｓｔ）＝’ｈｔｔｐ：／／ｕｒｉ’における．／ｔｅｓｔ）。

グループＢ：子孫のノードに対するアクセスの意味を含んだＸＰａｔｈ式。

Ｅｘ１：ｒｅｔｕｒｎに現れる要素のＸＰａｔｈ式（ある要素のｓｕｂｔｒｅｅなどを考える場合）。

Ｅｘ２：（様々なところに現れるだろう）テキスト値を参照する要素のＸＰａｔｈ式（／ｔｅｓｔ［．／ｘｍｌ＝’ＡＢＣ’］の／ｔｅｓｔ／ｘｍｌ）。

【００５５】

問い合わせ式から取り出された各パス式ｑに対して、検索オートマトン生成部

２１０にて検索オートマトンが生成され、アクセス制御オートマトン生成部２２０及びスキーマオートマトン生成部２３０にて生成された各オートマトンと共に、論理演算部２４０に入力されてアクセス権解析が行われ、以下のようにして、「常に許可」、「常に拒否」、「不確定」の判定が行われる。

・「常に拒否」の判定は、グループＡ、Ｂの別なくそのままパス式ｑに対して適用する。
・「常に許可」の判定は、パス式ｑがグループＡに属するならば、パス式ｑに対してそのまま適用し、パス式ｑがグループＢに属するならば、ｑ．Σ^*に変形してから適用する。
・「常に拒否」、「常に許可」のいずれの判定にも失敗したパス式ｑは、「不確定」であると判断される。

これらの判定結果は、例えば図１のメインメモリ１０３の作業領域に保持される。

【００５６】

上記のように判定する理由は次の通りである。

「グループＡのパス式ｑが常に拒否である」とは「パス式ｑの検索結果の全てのノードに対するアクセスが拒否である」ということであり、「グループＢのパス式ｑが常に拒否である」とは、「パス式ｑの検索結果の全てのノードに対するアクセスが拒否であり、その子孫ノードもまた見ることができない」という意味である。しかし、ＸＡＣＬによるアクセス制御ポリシーの記述では、所定のノードに対するアクセスが禁止であれば、その子孫も見ることはできないのであるから、「常に拒否」に関してはグループＡとグループＢとで差異はない。

一方、「常に許可」に関しては、「グループＡのパス式ｑが常に許可である」とは、「パス式ｑの検索結果の全てのノードに対するアクセスが許可される」ということである。これに対し、「グループＢのパス式ｑが常に許可である」とは、「パス式ｑの検索結果の全てのノードに対するアクセスが許可され、その子孫ノードもまた全て見ることができる」という意味となり、両者は異なる。グルー

プＢの場合は、全ての子孫ノードへのアクセスの許可を保障するため、ｑ．Σ^*で検索される全てのノードに対して、アクセス権の判定が成功しなければならない。したがって、上述したような判定手法を行うことが必要である。

図１４は、図１２の問い合わせ式から抽出された個々のパス式に対するアクセス権解析の結果（判定結果）を示す図表である。

【００５７】

問い合わせ式書き換え・判定部４６０は、図１４のように得られた個々のパス式に対する判定結果から問い合わせ式自体に対するアクセス可否の判定を行う。

問い合わせ式書き換え・判定部４６０は、まず、「常に拒否」と判断されたパス式に対して空集合を示す値（ここでは＄ｅｍｐｔｙとする）を代入する。

図１５は、図１２の問い合わせ式に対してかかる書き換えが行われた状態を示す図である。

この問い合わせ式に残っているパス式は、全て「常に許可」である。一般に、問い合わせ式に含まれる全てのパス式が「常に許可」か「常に拒否」であって「不確定」ではない場合、その問い合わせ式を「常に許可」であるパス式しか含まないように書き換えることができる。これは、「常に拒否」であるパス式については、そもそも当該パス式を用いたデータベース検索自体が無効となるからである。

したがって、このような問い合わせ式は、問い合わせ式書き換え・判定部４６０によって、「常に許可」であると判定され、実行時に動的なアクセス制御を行うことなく評価実行できる。問い合わせ式が「不確定」と判定されたパス式を含む場合は、問い合わせ式書き換え・判定部４６０によって、問い合わせ式自体が「不確定」と判定される。

以上のようにして、問い合わせ式書き換え・判定部４６０からは、必要に応じて書き換えられた問い合わせ式と、「常に許可」または「不確定」の判定結果とが出力される。

【００５８】

図１６は、本実施の形態によるアクセス権解析器４００の動作を説明するフローチャートである。

図１６に示すように、アクセス権解析器４００に処理対象である問い合わせ式が入力されると（ステップ１６０１）、まずパス式抽出部４５０により問い合わせ式中のパス式が抽出される（ステップ１６０２）。そして、検索オートマトン生成部２１０により検索オートマトンが生成され、アクセス制御オートマトン生成部２２０によりアクセス制御オートマトンが生成され、スキーマオートマトン生成部２３０によりスキーマオートマトンが生成される（ステップ１６０３）。

次に、生成された各オートマトンを用いて、論理演算部２４０により、問い合わせ式から抽出された個々のパス式のアクセス権が判定される（ステップ１６０４）。

【００５９】

次に、この個々のパス式のアクセス権判定結果に基づいて、問い合わせ式書き換え・判定部４６０により問い合わせ式のアクセス権が判定される。

まず、「常に拒否」と判定されたパス式がある場合は、そのパス式が空集合を示す値に書き換えられる（ステップ１６０５、１６０６）。その後、全てのパス式に対する判定結果に基づいて、問い合わせ式が「常に許可」または「不確定」と判定され、この判定結果が書き換えられた問い合わせ式と共に出力される（ステップ１６０７）。出力された判定結果及び問い合わせ式は、ＸＭＬデータベースに渡され、判定結果に応じてアクセス権解析が行われた後、もしくは直ちに当該問い合わせ式によるデータベース検索が行われる。

【００６０】

このように、本実施の形態によれば、ＸＭＬデータベース中のＸＭＬ文書に対して、データベース検索にかかる問い合わせ式のアクセス権が、少なくとも、「常に許可」、「不確定」のいずれに該当するかということが、当該ＸＭＬデータベースにおいて実際にＸＭＬ文書の構造を調べて行うアクセス権解析（ノード単位のアクセス権チェック）に先立って判断される。

「常に許可」である場合には、当該問い合わせ式によるデータベース検索では、ＸＭＬデータベースで、ＸＭＬ文書を調べて行う通常のアクセス権解析を行う必要はない。

したがって、本実施の形態による判定結果が「不確定」である場合にのみ、Ｘ

ＭＬ文書を調べて行う通常のアクセス権解析をＸＭＬデータベースにおいて実行すれば良く、データベース検索の処理全体における実行効率を向上させることができる。

【００６１】

なお、上記のように問い合わせ式を書き換えずに、その解釈実行において各パス式に対する「常に許可」、「常に拒否」の判定を利用することもできる。この場合、問い合わせ式の解釈実行システムは、パス式に対するアクセス権解析器の評価結果を組み合わせて実現されているものとする。このアクセス権解析器は、「常に許可」というパス式に対する検索結果を取り出すときには、データベースに対する実行時アクセス制御を行わず、また「常に拒否」というパス式に対しては、検索自体を行わないですむ。

【００６２】

本実施の形態では、アクセス権解析器４００は、実施の形態１のアクセス権解析器２００にパス式抽出部４５０及び問い合わせ式書き換え・判定部４６０を付加した構成について説明したが、スキーマオートマトンの生成を行わない実施の形態２のアクセス権解析器３００にパス式抽出部４５０及び問い合わせ式書き換え・判定部４６０を付加した構成とすることもできるのは言うまでもない。この場合、アクセス権解析器４００は、図８に示したように検索オートマトン生成部２１０、アクセス制御オートマトン生成部２２０、パステーブル管理部３３０及び論理演算部３４０を備え、さらにパス式抽出部４５０と問い合わせ式書き換え・判定部４６０とを備える。

【００６３】

アクセス権解析器４００の動作においても、上記ステップ１６０３において、スキーマオートマトンの生成は行われない。またステップ１６０４において、スキーマオートマトンの代わりにパステーブルを用いて、個々のパス式に対して「常に許可」、「常に拒否」、「不確定」の判定が行われるが、この動作については実施の形態２と同様である。その他の動作については上記と同様である。

【００６４】

〔実施の形態４〕

実施の形態４、５では、実施の形態１～３で説明したアクセス権解析器２００、３００、４００の実装例について説明する。実施の形態１～３では、アクセス制御を受けるサブジェクト（ｓｕｂｊｅｃｔ）が与えられた場合の静的な判定手法を説明したが、実際の実行環境では、アクセス制御のサブジェクトは動的に決定される。そこで、サブジェクトの特定を含め、データベース検索システムに実装されたアクセス権解析器について説明する。

問い合わせ式に対するアクセス権解析を実行するタイミングとしては、データベース検索の「実行直前」に行うシステムと、問い合わせ式の「コンパイル時」に行うシステムとが考えられる。実施の形態４では、「実行直前」に行うシステムを説明する。

【００６５】

図１７は、アクセス権解析を実行時に行うデータベース検索システムの構成例を示す図である。

図１７を参照すると、本実施の形態によるデータベース検索システムは、検索対象であるＸＭＬ文書を格納したＸＭＬデータベース１０と、検索条件を示す問い合わせ式（ＸＱｕｅｒｙ）を入力してパス式（ＸＰａｔｈ）を抽出するパス式抽出部２０と、パス式抽出部２０にて抽出されたパス式を用いてＸＭＬデータベース１０へのアクセス権限を判断するアクセス権解析器３０と、アクセス権解析器３０の解析結果に応じて問い合わせ式を書き換えると共に、当該問い合わせ式のアクセス権を判定する問い合わせ式書き換え・判定部４０とを備える。

このアクセス権解析器３０として、実施の形態１、２で説明したアクセス権解析器２００、３００を用いることができる。また、パス式抽出部２０は実施の形態３で説明したパス式抽出部４５０と同一であり、問い合わせ式書き換え・判定部４０は実施の形態３で説明した問い合わせ式書き換え・判定部４６０と同一である。したがって、アクセス権解析器３０と、パス式抽出部２０及び問い合わせ式書き換え・判定部４０とを合わせると、実施の形態３で示したアクセス権解析器４００の構成と等しい。

【００６６】

さて、データベース検索の実行直前にアクセス権解析を行う本実施の形態のシ

ステムの場合、以下のような特徴がある。

・パステーブルがある場合、最新の内容を参照することができる。
・サブジェクトが決定している。

したがって、パステーブルを用いてパス式のアクセス可否の判定を行う、実施の形態２で説明したアクセス権解析器３００を、本実施の形態のアクセス権解析器３０として用いることが好ましい。

【００６７】

図１７に示したデータベース検索システムにおいて、所定の問い合わせ式がパス式抽出部２０に入力されると、この問い合わせ式からパス式が抽出されて、アクセス権解析器３０に送られる。このとき、問い合わせ式による検索の主体であるサブジェクトも特定される。なお、問い合わせ式からパス式を抽出する処理は、問い合わせ式のコンパイル時に１回だけ行えば良い。そして、問い合わせ式と抽出されたパス式群を対応付けたテーブル（例えば図１３や図１４に示したテーブル）等を作成して管理しておくことにより、データベース検索を実行する度にパス式を抽出する煩雑さを避けられる。

アクセス権解析器３０で解析が行われた後、問い合わせ式書き換え・判定部４０によって、「常に拒否」と判断されたパス式の問い合わせ式における記述が書き換えられる。そして、「常に許可」もしくは「不確定」となった問い合わせ式がＸＭＬデータベース１０へ送られる。

ＸＭＬデータベース１０では、問い合わせ式書き換え・判定部４０での判定結果が「常に許可」である問い合わせ式に関しては、ＸＭＬ文書の構造を調べて行う通常のアクセス権解析を行うことなく、直ちにＸＭＬ文書の検索を実行する。また、問い合わせ式書き換え・判定部４０での判定結果が「不確定」である問い合わせ式に関しては、ＸＭＬ文書の構造を調べて行う通常のアクセス権解析を行った後に、ＸＭＬ文書の検索を実行する。

【００６８】

本実施の形態によるデータベース検索システムの場合、アクセス権解析をデー

タベース検索の実行時に行うため、問い合わせ式を発行してから検索結果を得るまでの処理にその分だけ時間を要することとなる。しかし、各パス式とサブジェクトとの組に対するアクセス権解析の結果をキャッシュしておくことによって、処理性能を向上させる（処理に要する時間を短縮する）ことができる。ただし、このキャッシュはパステーブルが更新されると無効になるので、そのたびにリセットすることが必要である。

【００６９】

〔実施の形態５〕

実施の形態５では、問い合わせ式の「コンパイル時」にアクセス権解析を行うシステムを説明する。

図１８は、アクセス権解析を実行時に行うデータベース検索システムの構成例を示す図である。

図１８を参照すると、本実施の形態によるデータベース検索システムは、検索対象であるＸＭＬ文書を格納したＸＭＬデータベース１０と、検索条件を示す問い合わせ式（ＸＱｕｅｒｙ）を入力してパス式（ＸＰａｔｈ）を抽出するパス式抽出部２０と、パス式抽出部２０にて抽出されたパス式を用いてＸＭＬデータベース１０へのアクセス権限を判断するアクセス権解析器３０と、アクセス権解析器３０の解析結果に応じて問い合わせ式を書き換えると共に、当該問い合わせ式のアクセス権を判定する問い合わせ式書き換え・判定部４０と、問い合わせ式書き換え・判定部４０による判定結果を保持する記憶部５０とを備える。

このアクセス権解析器３０として、実施の形態１、２で説明したアクセス権解析器２００、３００を用いることができる。また、アクセス権解析器３０と、パス式抽出部２０及び問い合わせ式書き換え・判定部４０とを合わせると、実施の形態３で示したアクセス権解析器４００の構成と等しいことは、実施の形態４の場合と同様である。

記憶部５０は、キャッシュメモリであり、例えばデータベース検索システムを図１に示したハードウェアで構成した場合、メインメモリ１０３で実現される。

【００７０】

さて、問い合わせ式のコンパイル時に予めアクセス権解析を行う本実施の形態

のシステムの場合、以下のような特徴がある。

・アクセス権解析後にもパステーブルが更新され得るため、パステーブルを用いることができない。
・サブジェクトは決定していない。

したがって、アクセス可否の判定にパステーブルを用いない実施の形態１で説明したアクセス権解析器２００を、本実施の形態のアクセス権解析器３０として用いることが好ましい。

【００７１】

図１８に示したデータベース検索システムにおいて、所定の問い合わせ式がパス式抽出部２０に入力されると、この問い合わせ式からパス式が抽出されて、アクセス権解析器３０に送られる。この時点ではサブジェクトは特定されていないため、可能な様々なサブジェクトを想定してアクセス権解析器３０によるアクセス権解析が行われる。

アクセス権解析器３０で解析が行われた後、問い合わせ式書き換え・判定部４０によって、「常に拒否」と判断されたパス式の問い合わせ式における記述が書き換えられる。そして、「常に許可」もしくは「不確定」となった問い合わせ式が記憶部５０へ送られ、キャッシュされる。

データベース検索の実行時には、記憶部５０にキャッシュされているアクセス権解析結果を用いて問い合わせ式のアクセス可否が認識される。そして、判定結果が「常に許可」である問い合わせ式に関しては、ＸＭＬ文書の構造を調べて行う通常のアクセス権解析を行うことなく、直ちにＸＭＬ文書の検索を実行する。また、問い合わせ式書き換え・判定部４０での判定結果が「不確定」である問い合わせ式に関しては、ＸＭＬ文書の構造を調べて行う通常のアクセス権解析を行った後に、ＸＭＬ文書の検索を実行する。なお、所定のサブジェクトから発行された所定の問い合わせがキャッシュにヒットしない場合には、当該問い合わせ式のコンパイルをやり直す。

【００７２】

本実施の形態によるデータベース検索システムの場合、アクセス権解析を問い合わせ式のコンパイル時に行うため、データベース検索の実行時には記憶部５０にキャッシュされた判定結果のみを参照することとなり、高速な処理を実現することが可能である。しかしながら、実施の形態４と比較すると、予め可能なサブジェクトを想定して、その全ての場合の判定結果を記憶部５０に保持しておかなければならないため、記憶部５０を実現する記憶手段における記憶容量の使用量が大きい。

【００７３】

なお、上述した各実施の形態では、検索オートマトン、アクセス制御オートマトン、スキーマオートマトンにおいて、パスのみを扱っている。パスは単なる列であり、普通のオートマトンによる取り扱いが簡単であるという長所を持つ。しかし、ＸＭＬ文書の構造は木構造であり、スキーマはＸＭＬ文書がどんな木構造を許容するかを表している。アクセス制御ポリシーの記述に用いるＸＰａｔｈ検索式も、一般的には木構造に関するものであり、パスだけを扱うものではない。したがって、上記各実施の形態で示した列を扱うオートマトンではなく、木構造を扱うオートマトンを用いて上述したアクセス権解析器２００、３００、４００、３０を実現することも可能である。

【００７４】

また、上記各実施の形態では、ＸＭＬ文書を処理対象として、ＸＭＬデータベースの検索におけるアクセス権解析について説明したが、データベース検索以外の場面でも、広くＸＰａｔｈ等のパス式を条件とするアクセス権限の解析を行う場合に、上記各実施の形態を適用できるのは言うまでもない。

さらに、ＸＭＬ以外の構造化文書を対象とするシステムにおいても、スキーマやアクセス制御ポリシーに相当する仕様が定義されているならば、上記各実施の形態を適用することが可能である。

【００７５】

【発明の効果】

以上説明したように、本発明によれば、ＸＭＬデータベースにおいて、ＸＭＬ文書自体やそのノードを調べることなくアクセス権解析を行うことを実現し、Ｘ

ＭＬデータベースの検索性能を向上させることができる。

【図面の簡単な説明】

【図１】　本実施の形態によるアクセス権解析器を実現するのに好適なコンピュータ装置のハードウェア構成の例を模式的に示した図である。

【図２】　実施の形態１によるアクセス権解析器の機能構成を示すブロック図である。

【図３】　本実施の形態の検索対象となるＸＭＬ文書の例を示す図である。

【図４】　図３のＸＭＬ文書に対して、検索したいＸＰａｔｈ式が全てのａｕｔｈｏｒ要素である場合の検索オートマトンを示す図である。

【図５】　図３のＸＭＬ文書に対して、検索したいＸＰａｔｈ式がｂｉｂ要素の下のｂｏｏｋ要素の全ての子供である場合の検索オートマトンを示す図である。

【図６】　本実施の形態におけるアクセス制御オートマトンを説明する図である。

【図７】　本実施の形態における「常に拒否」の判定規則を説明する図である。

【図８】　実施の形態２によるアクセス権解析器の機能構成を示すブロック図である。

【図９】　図３に示したＸＭＬ文書に対して作成されるパステーブルの例を示す図である。

【図１０】　図９のパステーブルにアクセス制御の判定情報を登録した様子を示す図である。

【図１１】　実施の形態３によるアクセス権解析器の機能構成を示すブロック図である。

【図１２】　問い合わせ式（ＸＱｕｅｒｙ）の例を示す図である。

【図１３】　図１２の問い合わせ式から抽出されるパス式の一覧を示す図表である。

【図１４】　図１２の問い合わせ式から抽出された個々のパス式に対するアクセス権解析の結果（判定結果）を示す図表である。

【図１５】　図１２の問い合わせ式に対してかかる書き換えが行われた状態を示す図である。

【図１６】　本実施の形態によるアクセス権解析器の動作を説明するフローチャートである。

【図１７】　実施の形態４によるアクセス権解析を実行時に行うデータベース検索システムの構成例を示す図である。

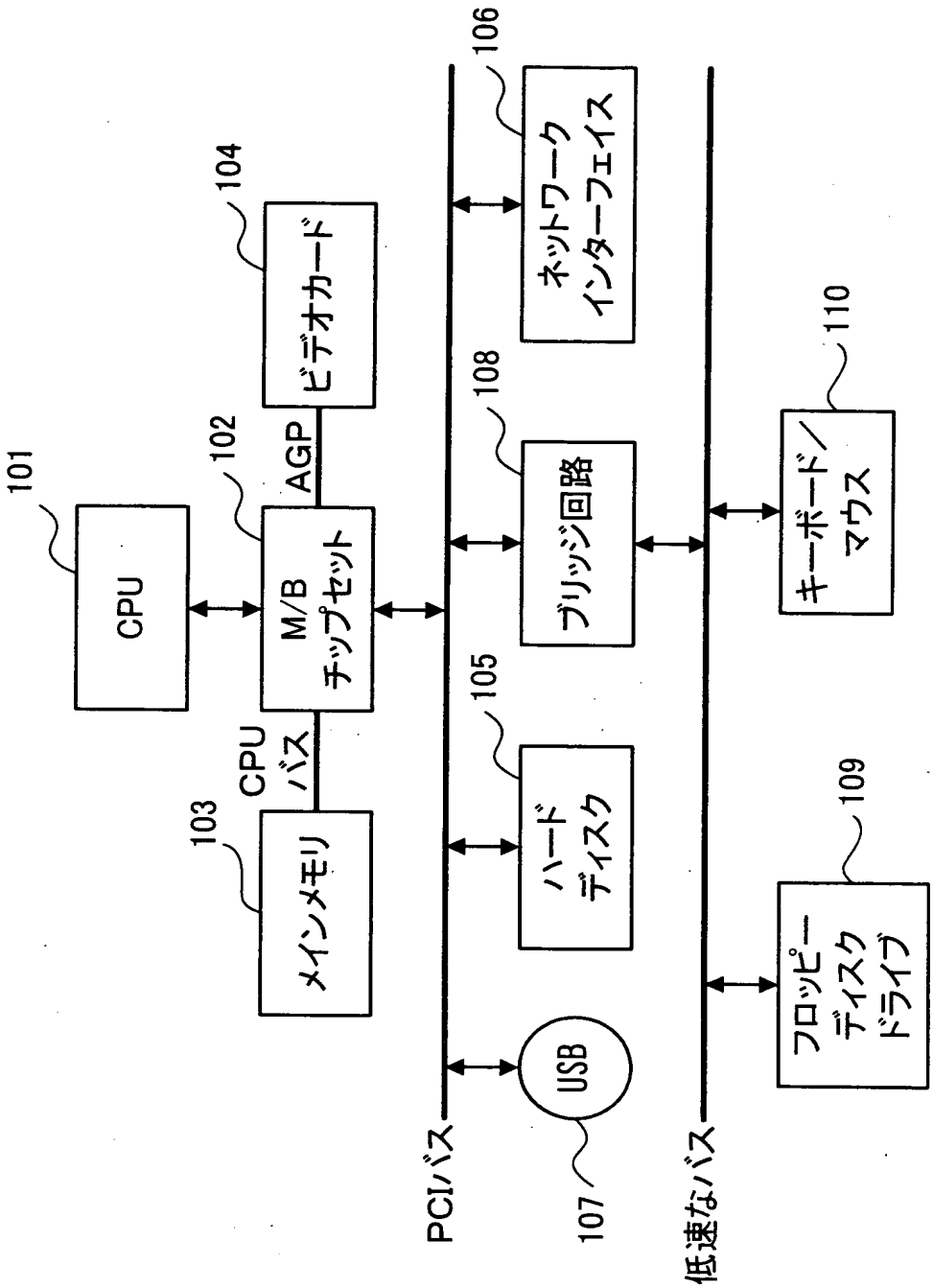
【図１８】　実施の形態５によるアクセス権解析を実行時に行うデータベース検索システムの構成例を示すブロック図である。

【符号の説明】

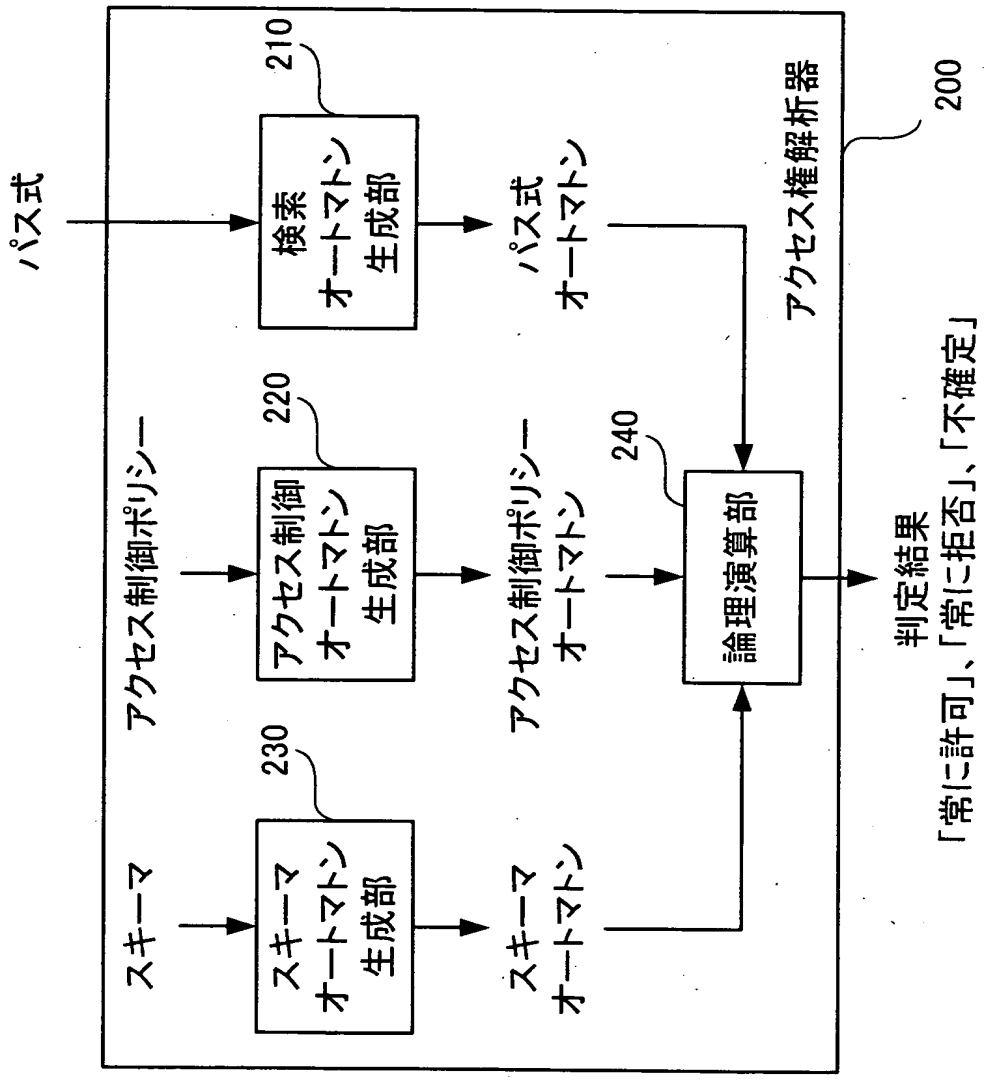
１０…ＸＭＬデータベース、２０、４５０…パス式抽出部、３０、２００、３００、４００…アクセス権解析器、４０、４６０…問い合わせ式書き換え・判定部、２１０…検索オートマトン生成部、２２０…アクセス制御オートマトン生成部、２３０…スキーマオートマトン生成部、２４０、３４０…論理演算部、３３０…パステーブル管理部、３３１…パステーブル、１０１…ＣＰＵ、１０２…Ｍ／Ｂチップセット、１０３…メインメモリ、１０５…ハードディスク

【書類名】　　図面

【図１】


CPU
101
M/B
チップセット
102
AGP
ビデオカード
104
CPU
バス
メインメモリ
103
PCIバス
ハード
ディスク
105
USB
107
ブリッジ回路
108
ネットワーク
インターフェイス
106
低速なバス
キーボード／
マウス
110
フロッピー
ディスク
ドライブ
109

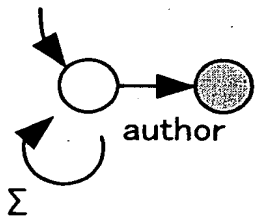
【図２】


パス式
検索
オートマトン
生成部
210
パス式
オートマトン
アクセス権解析器
200
アクセス制御ポリシー
アクセス制御
オートマトン
生成部
220
アクセス制御ポリシー
オートマトン
240
論理演算部
判定結果
「常に許可」、「常に拒否」、「不確定」
スキーマ
スキーマ
オートマトン
生成部
230
スキーマ
オートマトン

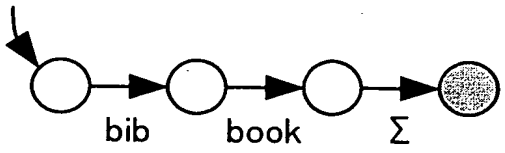
【図３】

```
<bib>
   <book>
       <title>TCP/IP illustrated</title>
       <author>W. Stevens</author>
       <price>50.00</price>
   </book>
   <book>
       <title>Data on the Web</title>
       <author>S. Abiteboul</author>
       <author>P. Buneman</author>
       <author>D. Suciu</author>
       <price>40.00</price>
   </book>
</bib>
```

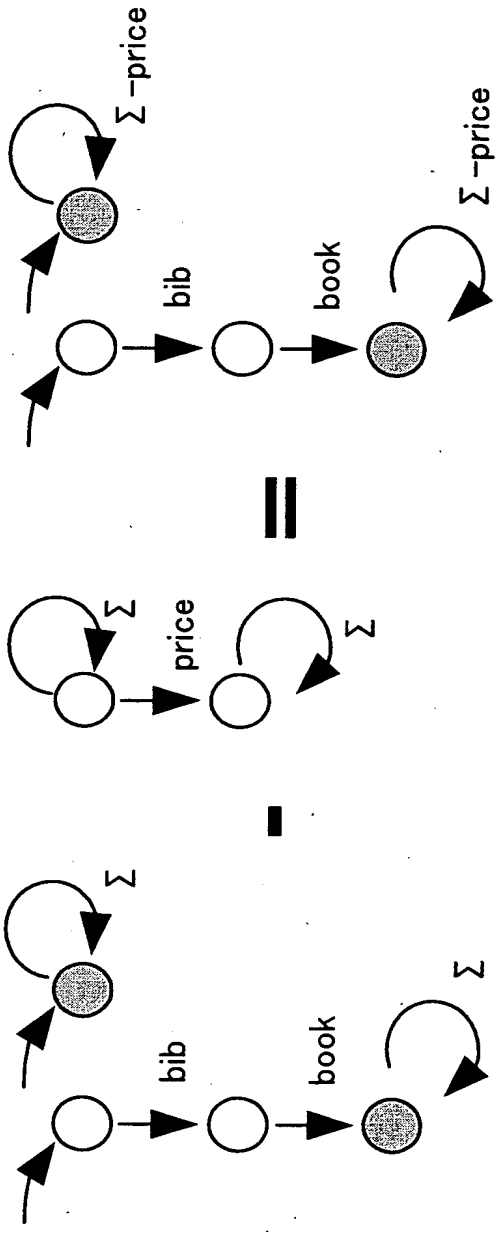
【図４】



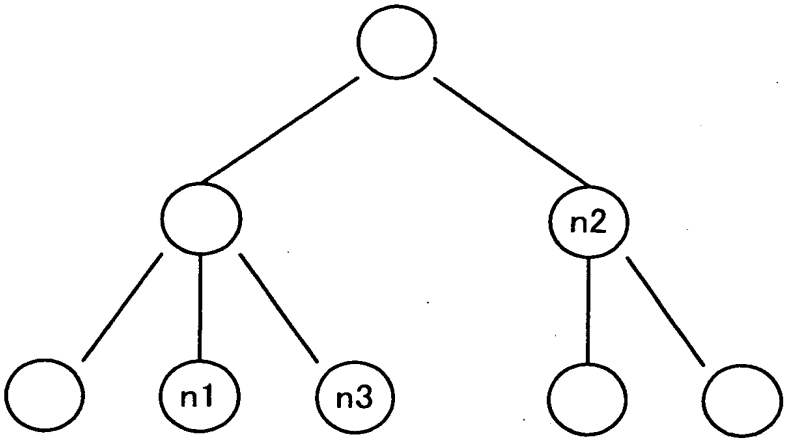
【図５】



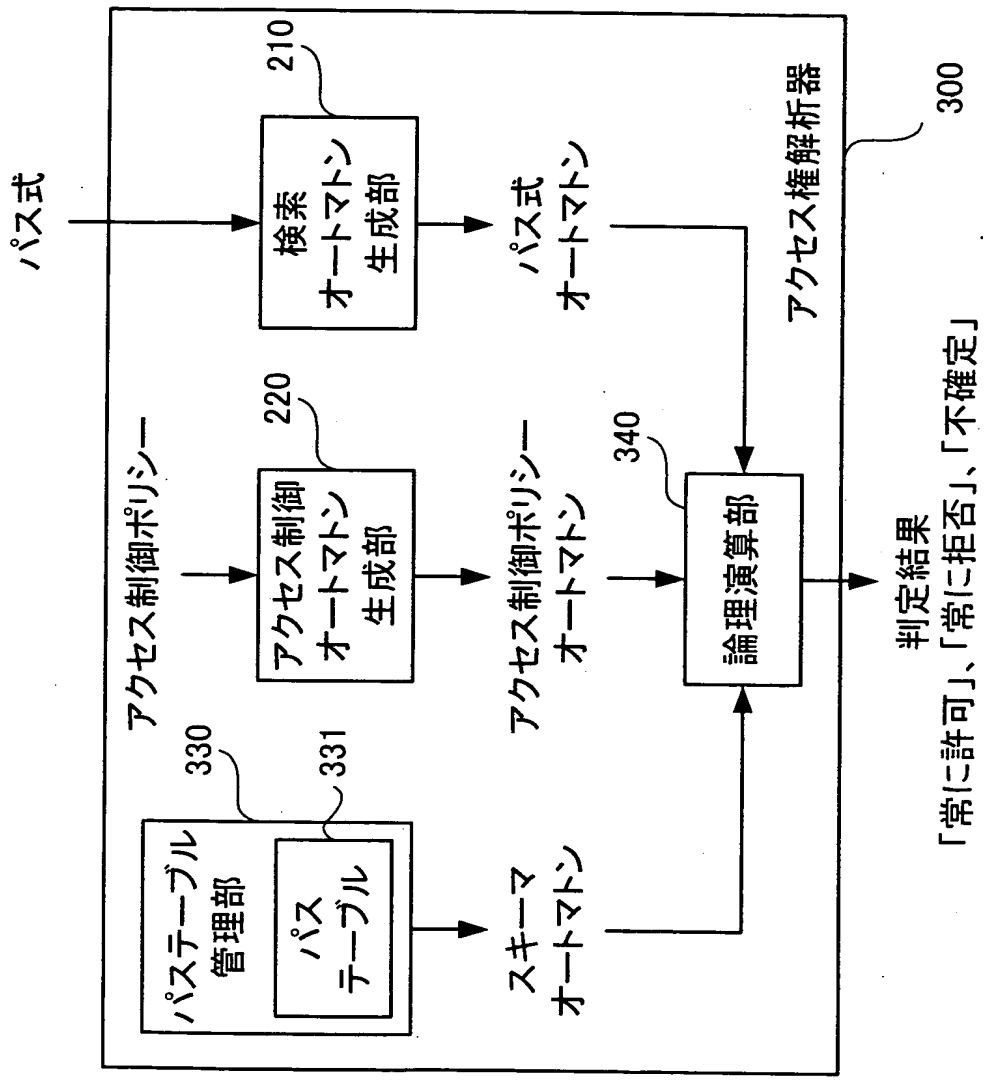
【図６】



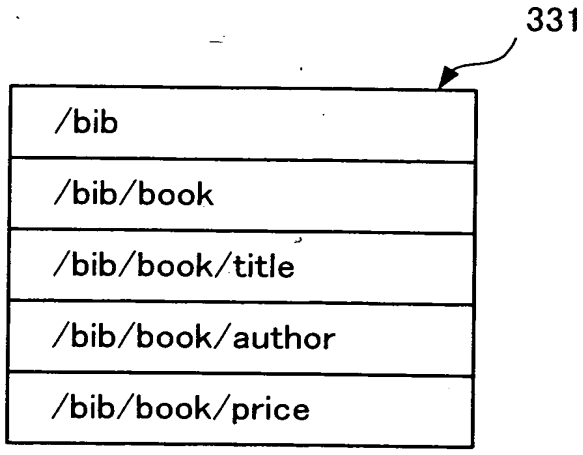
【図７】



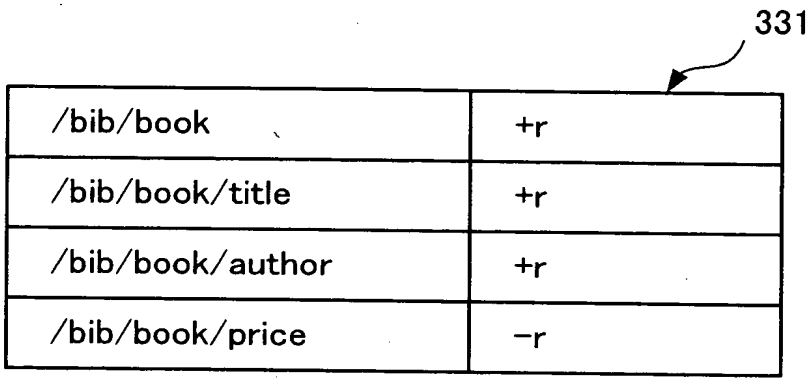
【図８】


パス式
検索
オートマトン
生成部
210
パス式
オートマトン
アクセス権解析器
300
アクセス制御ポリシー
アクセス制御
オートマトン
生成部
220
アクセス制御ポリシー
オートマトン
340
論理演算部
判定結果
「常に許可」、「常に拒否」、「不確定」
330
パステーブル
管理部
331
パス
テーブル
スキーマ
オートマトン

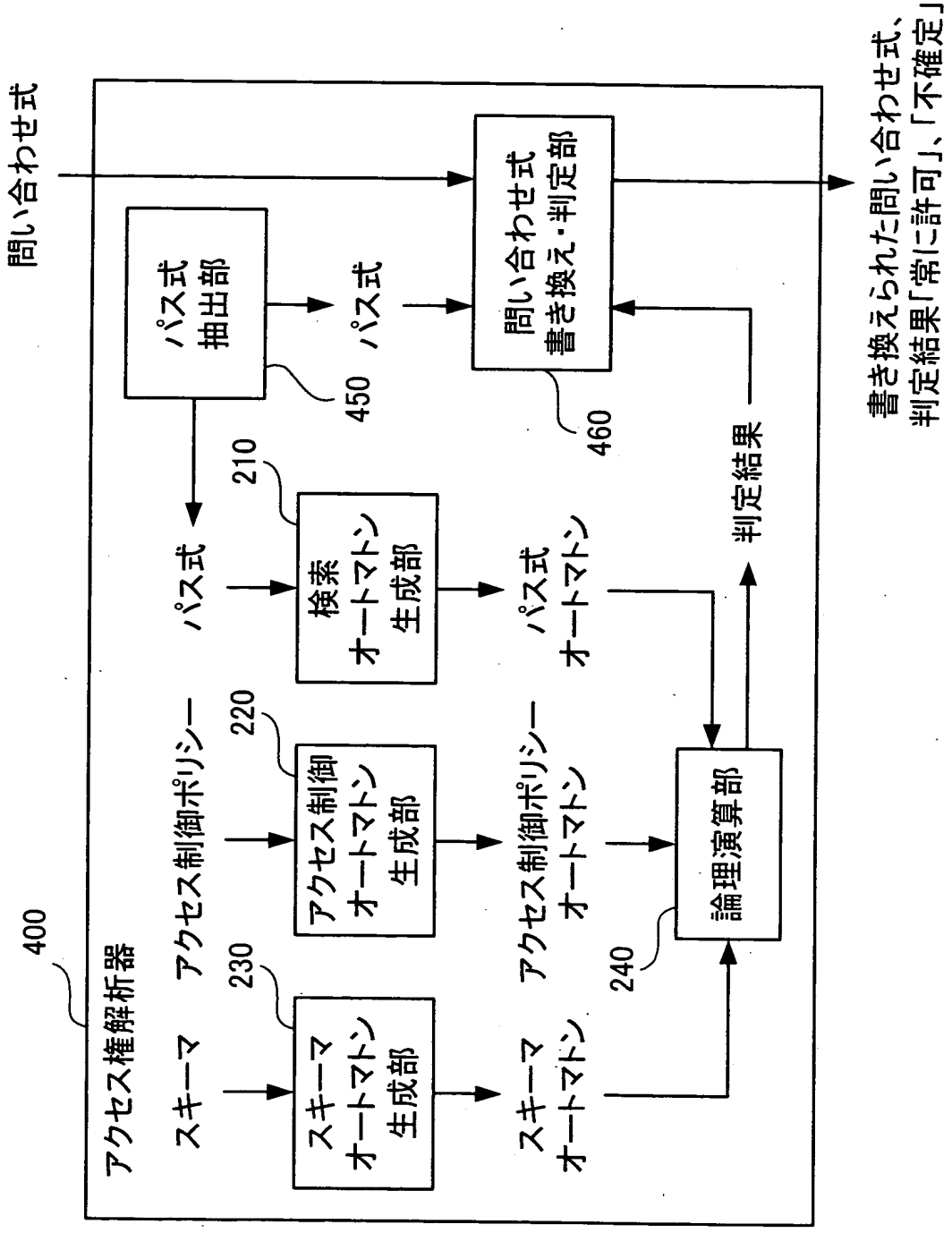
【図９】


331
/bib
/bib/book
/bib/book/title
/bib/book/author
/bib/book/price

【図１０】


331
/bib/book
+r
/bib/book/title
+r
/bib/book/author
+r
/bib/book/price
−r

【図１１】


問い合わせ式
パス式
抽出部
450
パス式
問い合わせ式
書き換え・判定部
460
書き換えられた問い合わせ式、
判定結果「常に許可」、「不確定」
判定結果
400
アクセス権解析器
スキーマ
230
スキーマ
オートマトン
生成部
スキーマ
オートマトン
アクセス制御ポリシー
220
アクセス制御
オートマトン
生成部
アクセス制御ポリシー
オートマトン
パス式
210
検索
オートマトン
生成部
パス式
オートマトン
240
論理演算部

【図１２】

```
1. <bib> {
2.  for $b in document("http://www.bn.com")/bib/book
3.  where $b/publisher = "Addison-Wesley" and $b/@year > 1991
4.  return
5.    <book year={ $b/@year }>
6.     { $b/title }
7.    </book>
8. }
9. </bib>
```

【図１３】

問い合わせ式上の表現	取り出されたパス式(XPath)	グループ
２行目，document .. /book	document ("http://www.bn.com") /bib/book	A
３行目，$b/publisher	document ("http://www.bn.com") /bib/book/publisher	B
３行目，$b/@year	document ("http://www.bn.com") /bib/book/@year	B
５行目，$b/@year	document ("http://www.bn.com") /bib/book/@year	B
６行目，$b/title	document ("http://www.bn.com") /bib/book/title	B

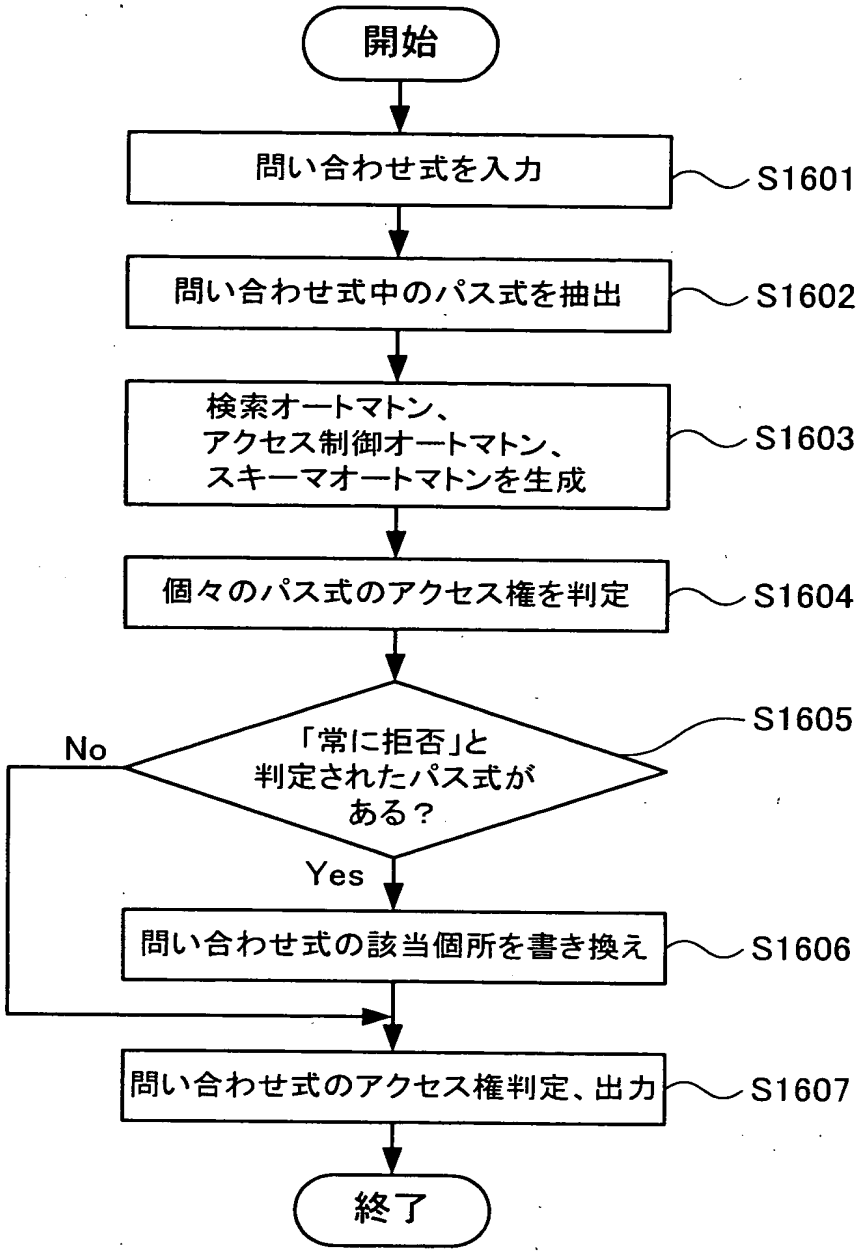
【図１４】

問い合わせ式上の表現	取り出されたパス式(XPath)	解析結果
2行目，document .. /book	document("http://www.bn.com")/bib/book	常に許可
3行目，$b/publisher	document("http://www.bn.com")/bib/book/publisher	常に許可
3行目，$b/@year	document("http://www.bn.com")/bib/book/@year	常に拒否
5行目，$b/@year	document("http://www.bn.com")/bib/book/@year	常に拒否
6行目，$b/title	document("http://www.bn.com")/bib/book/title	常に許可

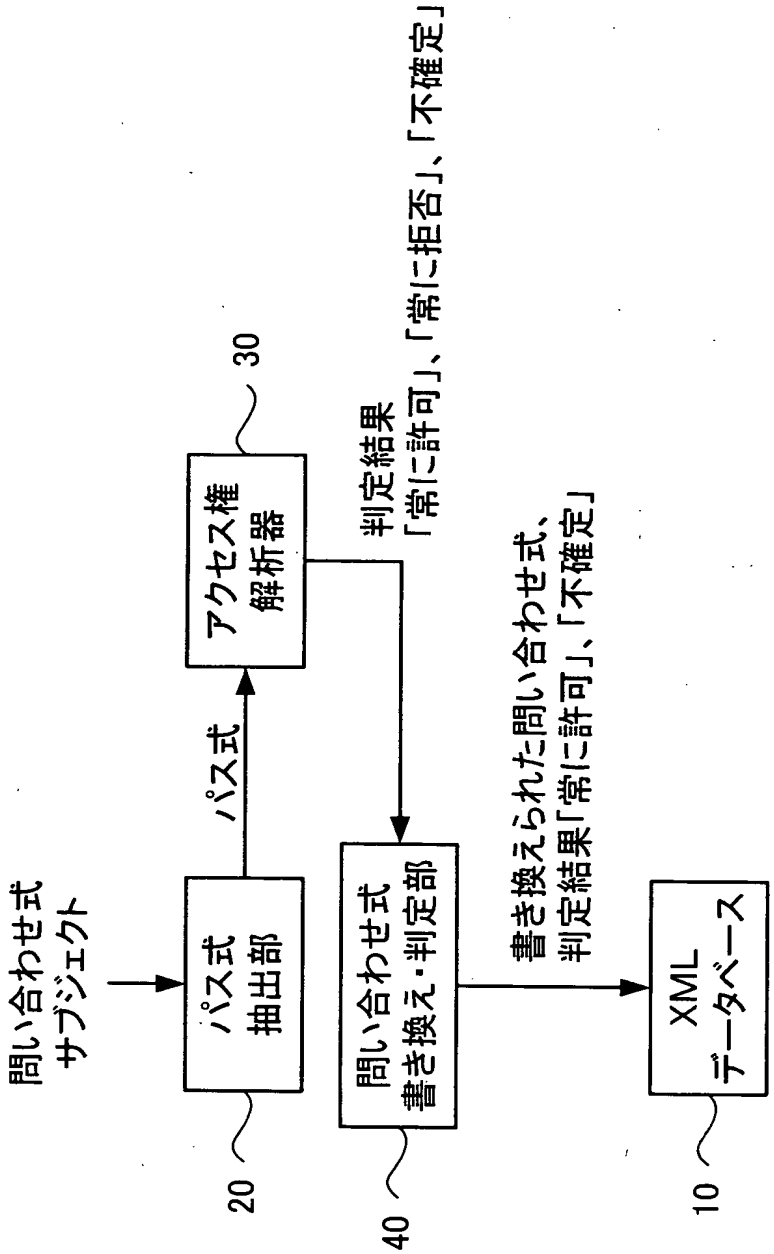
【図１５】

```
1. <bib> {
2.   for $b in document("http://www.bn.com")/bib/book
3.   where $b/publisher = "Addison-Wesley" and $empty > 1991
4.   return
5.     <book year={ $empty }>
6.       { $b/title }
7.     </book>
8. }
9. </bib>
```

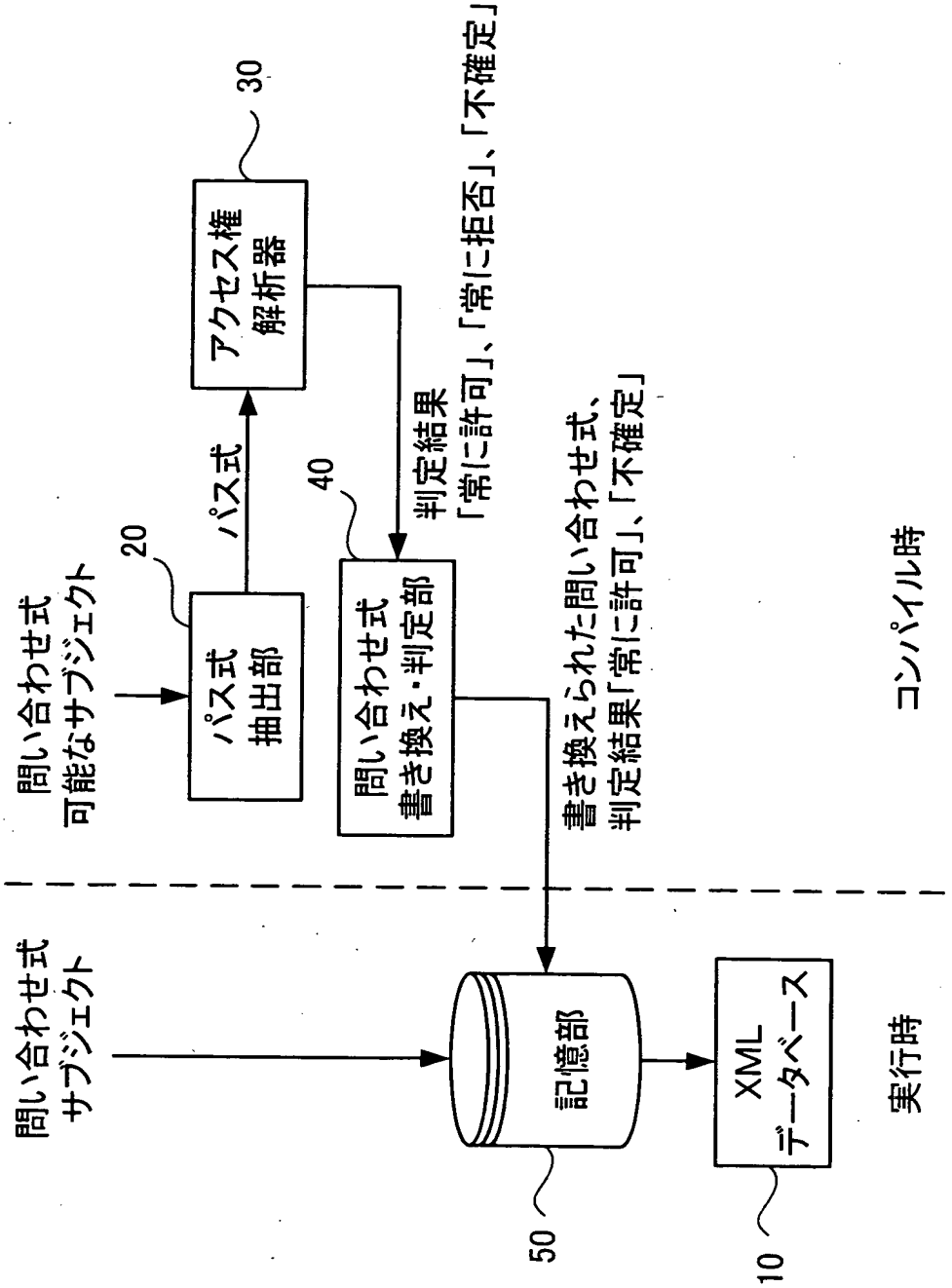
【図１６】


開始
問い合わせ式を入力
S1601
問い合わせ式中のパス式を抽出
S1602
検索オートマトン、
アクセス制御オートマトン、
スキーマオートマトンを生成
S1603
個々のパス式のアクセス権を判定
S1604
「常に拒否」と
判定されたパス式が
ある？
S1605
No
Yes
問い合わせ式の該当個所を書き換え
S1606
問い合わせ式のアクセス権判定、出力
S1607
終了

【図１７】


問い合わせ式
サブジェクト
パス式
抽出部
20
パス式
アクセス権
解析器
30
判定結果
「常に許可」、「常に拒否」、「不確定」
問い合わせ式
書き換え・判定部
40
書き換えられた問い合わせ式、
判定結果「常に許可」、「不確定」
XML
データベース
10

【図１８】


問い合わせ式
可能なサブジェクト
20
パス式
抽出部
パス式
30
アクセス権
解析器
40
問い合わせ式
書き換え・判定部
判定結果
「常に許可」、「常に拒否」、「不確定」
書き換えられた問い合わせ式、
判定結果「常に許可」、「不確定」
コンパイル時
問い合わせ式
サブジェクト
記憶部
50
XML
データベース
10
実行時

【書類名】　　要約書

【要約】

【課題】　ＸＭＬデータベースにおいて、ＸＭＬ文書自体やそのノードを調べることなくアクセス権解析を行うことを実現し、ＸＭＬデータベースの検索性能を向上させる。

【解決手段】　データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部２１０と、アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部２２０と、スキーマからスキーマオートマトンを生成するスキーマオートマトン生成部２３０と、生成された各オートマトンに関する論理演算を行う論理演算部２４０とを備える。これによって、ＸＭＬデータベースにおいてＸＭＬ文書自体を調べ、ノードごとのチェックを行うことなく、かかるパス式を用いたデータベース検索におけるアクセス権を判定する。

【選択図】　　図２

認定・付加情報

特許出願の番号	特願２００３－０８３２４３
受付番号	５０３００４８３３８３
書類名	特許願
担当官	小野寺　光子　　１７２１
作成日	平成１５年　５月　６日

＜認定情報・付加情報＞

【特許出願人】
- 【識別番号】 390009531
- 【住所又は居所】 アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
- 【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】
- 【識別番号】 100086243
- 【住所又は居所】 神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内
- 【氏名又は名称】 坂口　博

【代理人】
- 【識別番号】 100091568
- 【住所又は居所】 神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内
- 【氏名又は名称】 市位　嘉宏

【代理人】
- 【識別番号】 100108501
- 【住所又は居所】 神奈川県大和市下鶴間１６２３番１４　日本アイ・ビー・エム株式会社　知的所有権
- 【氏名又は名称】 上野　剛史

【復代理人】 申請人
- 【識別番号】 100104880
- 【住所又は居所】 東京都港区赤坂５－４－１１　山口建設第２ビル６Ｆ　セリオ国際特許事務所
- 【氏名又は名称】 古部　次郎

【選任した復代理人】
- 【識別番号】 100118201

次頁有

認定・付加情報（続き）

【住所又は居所】　東京都港区赤坂５－４－１１　山口建設第二ビル
６Ｆ　セリオ国際特許事務所
【氏名又は名称】　千田　武

次頁無

出　願　人　履　歴　情　報

識別番号　　〔３９０００９５３１〕

１．変更年月日　　２００２年　６月　３日
［変更理由］　住所変更
住　所　　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
氏　名　　インターナショナル・ビジネス・マシーンズ・コーポレーション